Adaptive Motion Planning Based on Vehicle Characteristics and Regulations for Off-Road UGVs

Yonghoon Ji , *Member, IEEE*, Yusuke Tanaka, *Student Member, IEEE*,
Yusuke Tamura , *Member, IEEE*, Mai Kimura, Atsushi Umemura, Yoshiharu Kaneshima,
Hiroki Murakami, Atsushi Yamashita , *Member, IEEE*, and Hajime Asama , *Fellow, IEEE*

Abstract—In this paper, we propose a novel motion planning method for off-road unmanned ground vehicles, based on three-dimensional (3-D) terrain map information. Previous studies on the motion planning of a vehicle traveling on rough terrain dealt only with a relatively small environment. Furthermore, unique vehicle characteristics were not considered, and it was also impossible to incorporate regulations, such as maintaining driving speed and suppressing posture change. The proposed method enables vehicles to adaptively generate a path by considering vehicle characteristics and the regulations, in a large-scale environment, with rough terrain. A random sampling based scheme was applied to carrying out global path planning, based on a 3-D environmental model. Experimental results showed that the proposed off-road motion planner could generate an appropriate path, which satisfies vehicle characteristics and predefined regulations.

Index Terms—Autonomous navigation, field robots, motion planning, unmanned ground vehicle (UGV).

I. INTRODUCTION

IN RECENT years, autonomous mobile robots and unmanned ground vehicles (UGVs) have attracted the attention of many researchers, and are becoming capable of dealing with various environments. Safe and reliable motion planning for mobility

Y. Ji is with the Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University, Tokyo 112-8551, Japan (e-mail: ji@mech.chuo-u.ac.jp).

Y. Tanaka is with the Toyota Motor Corporation, Toyota 471-8571, Japan (e-mail: yusuke_tanaka_ab@mail.toyota.co.jp).

Y. Tamura, A. Yamashita, and H. Asama are with the Department of Precision Engineering, School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: tamura@robot.t.u-tokyo.ac.jp; yamashita@robot.t.u-tokyo.ac.jp; asama@robot.t.u-tokyo.ac.jp).

M. Kimura, Y. Kaneshima, and H. Murakami are with the IHI Corporation, Yokohama 235-8501, Japan (e-mail: mai_kimura@ihi.co.jp; yoshiharu_kaneshima@ihi.co.jp; hiroki_murakami@ihi.co.jp).

A. Umemura is with the Silicon Studio Corporation, Tokyo 150-0013, Japan (e-mail: atsushi.umemura@siliconstudio.co.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2870662

is one of the most important requirements for such unmanned robots. In case of indoor service robots who interact with humans, considerable studies to solve motion planning problem have been intensively conducted and the reliable theories have already been established [1]–[4]. However, there have been very few studies on establishing an outdoor motion planning methodology in off-load environments, despite the undeniable fact that it is an indispensable requirement to operate unmanned robots traveling on rough terrain such as disaster sites where hazards prohibit human access.

When the UGV navigates autonomously on an off-road environment where rough terrain exists, the importance of avoiding accidents, such as collision and turnover, cannot be overemphasized in the sense of safe navigation. Therefore, the UGV is required to avoid such risks and to select a route within the traversable area. For the UGV to navigate safely on rough terrain, it is necessary to estimate the traversability for rough terrain using sensor data and to perform appropriate motion planning. As a method for estimating traversability on rough terrain, Hata and Wolf [5] employed an approach, which defines a two-class classification problem based on the support vector machine (SVM). They extracted terrain features as input values of the SVM classifier, from the target area, by using distance and intensity data acquired from the laser range finder (LRF). In this approach, training data should be labeled manually by human instruction, which is necessary to learn the classifier. Therefore, sufficient performance may not be produced in the case where it is very difficult to carry out proper labeling. In this respect, Tanaka *et al.* [6] proposed a method that can perform terrain traversability analysis (TTA) on rough terrain, without a training dataset. They extracted three kinds of features consisting of unevenness and inclination information, with respect to the area surrounding the mobile robot. Then, traversability was calculated based on fuzzy inference, which constitutes the membership function, through the value of these features. Proposals by Lalonde *et al.* [7] and Bellone *et al.* [8] applied principal component analysis (PCA) to point cloud data that correspond to a surrounding environment to perform the terrain classification tasks. From the PCA results containing point cloud's geometric shape, inclination, and unevenness, the traversability of the surrounding environment can be analyzed. However, the purpose of these approaches is to only estimate the traversability of rough terrain. Thus, no methodology of the UGV's motion

planning based on the TTA result has yet been proposed. To solve the motion planning problem of an UGV, with respect to rough terrain, an appropriate motion panning scheme based on TTA results, should be taken into consideration. The following section introduces previous studies dealing with the motion planning problem with regard to rough terrain navigation for UGVs. Regarding the UGV's motion planning schemes, nowadays random sampling based methodologies, such as rapidly exploring random trees (RRT) [9], probabilistic roadmap method (PRM) [10], and path-directed subdivision trees (PDST) [11], are most widely used, due to computational power having improved rapidly over the past decades.

At the defense advanced research projects agency (DARPA) grand challenge held a decade ago, one of the main problems was autonomous navigation of an UGV on a highly controlled desert area with rough terrain [12], [13]. Many teams applied the random sampling based motion planning approach, so that some breakthroughs have been made for the off-road UGV. In this challenge, a global path was given in advance and each vehicle performs local motion planning while identifying road conditions and obstacles. In real situations, such as a disaster site, however, there is almost no case in which the entire global path is pregiven. Furthermore, most other studies related to rough terrain navigation have the limitations described ahead. First, there has been no study that considered the unique characteristics of each vehicle. Second, previous studies did not incorporate regulations necessary for UGV operation, depending on different situations, such as maintaining driving speed and suppressing posture change. In Section III, vehicle characteristics, and the regulations defined in this paper, are described in detail. Finally, previous studies dealt only with a relatively small environment, owing to computational load limitations.

To overcome the limitations mentioned before, we propose an adaptive methodology for global motion planning by considering all degrees of freedom (DoFs) of vehicle pose in a pregiven relatively large-scale environment map including rough terrain. Here, "adaptive" means that the proposed methodology enables to perform appropriate planning that satisfies different conditions defined from the vehicle characteristics and the regulations. To achieve the adaptive motion planning according to the unique characteristics of each vehicle, we propose a novel method, which defines configuration obstacles to take the motion model of a vehicle, and terrain information into consideration, when expending the tree structure for motion planning. In addition, a novel method for generating control velocity using biased sampling is proposed, to perform the adaptive motion planning by considering regulations, which is one of the crucial objectives in this study. Finally, using our proposed motion planner, we can identify a solution within feasible processing times, even in a relatively large-scale environment compared to environments covered by previous studies.

The remainder of this paper is organized as follows. After the discussion of relevant studies in Section II, we explicitly state the problem and the purpose in Section III. Section IV introduces the definitions of the vehicular characteristics and regulations, which are the main focal points of this study. Section V presents the overall structure of the proposed motion planner. Section VI briefly summarizes the random sampling based motion planner. Section VII presents the proposed tree expansion scheme in

the random sampling process and a detailed description for configuration obstacle design is presented in Section VIII. Then, in Section IX, the effectiveness of the proposed motion planner for an off-road UGV is evaluated by the experimental results discussed in Section IV. Finally, Section X concludes this paper and discusses future work.

II. RELATED RESEARCH

There are several studies that solve the problem of motion planning or path planning when vehicles navigate safely from the initial position to the destination. Among them, regarding the vehicles traveling on rough terrain, very few methods were developed to solve the motion planning problem for a planetary rover or UGV.

Ono *et al.* [14] applied a random sampling based path planning method to the planetary rover. In this method, the traversability of rough terrain surrounding the vehicle was analyzed based on captured images from a stereo camera mounted on a vehicle and the estimation result of the traversability was mapped to digital elevation model. With respect to the planning algorithm, they combined the A* search algorithm with a rapidly exploring random graph (RRG) [15], which is one of the random sampling based methods. As a result, a safe path could be generated among the areas estimated to be traversable. Lee *et al.* [16] employed an RRT-based approach for the mobile robot traveling on rough terrain. This approach adopted a two-dimensional (2-D) grid map as an environment model, and a cost value estimated from TTA was assigned to each grid. To generate a tree structure consisting of random nodes, a heuristic-based node selection method, which uses cost values, was proposed in this study. In this method, even though the experiment was conducted in a relatively large-scale environment map of 400 m^2, a path solution to the destination was generated with very short processing time of approximately 5 s. Kuwata *et al.* [17] also applied an RRT-based approach to the path planning of the UGV. In this approach, a bias was given to the sampling region, depending on the vehicle's situation when generating random nodes; therefore, it was possible to prevent the generation of random nodes in the area which could not, obviously, become the path solution in advance. For example, in a situation where the mobile robot turned right, random nodes were generated intensively to the right-hand side of the vehicle, whereas sampling to an unnecessary region, including the left-hand side of the vehicle, was suppressed, so that the vehicle could efficiently determine the path for the right turn. Richter *et al.* [18] proposed a method of an online motion planning algorithm, which simultaneously performs self-localization and map building using sensor measurement data for the target environment of the UGVs. By this method, a high risk value was assigned to the unknown area, where the environment map was not built because of various factors, such as occlusion. Regarding the assignment of a risk value to an unknown area, they obtained heuristics for various situations, through preliminary learning in the simulation environment. Therefore, it was possible to suppress the generation of the vehicle's motion in an uncertain region, such as the occlusion area. Fassbender *et al.* [19] developed a system for UGVs, which simultaneously carries out appropriate traversability estimation and motion planning, even in convoluted environ-

ments such as a forest. The system also independently performs self-localization and mapping, similar to the proposal by Richter *et al.*, to reflect the results of traversability estimation as the conditions when searching for a solution. Feasible motion, with consideration to vehicle speed, could be generated as a result. However, the above-mentioned proposals by Ono *et al.*, Lee *et al.*, Kuwata *et al.*, Richter *et al.*, and Fassbender *et al.* have similar limitations. For instance, path search is performed under the assumption that the motion of the vehicle is defined in a 2-D plane; thus, it cannot manage the 3-D information (i.e., difference in elevation or posture information) of the vehicle, which must be considered in rough terrain. In addition, it cannot produce adaptive path reflecting vehicle characteristics, such as driving speed and hill-climbing performance (i.e., maximum allowable angle of inclination for a vehicle, since these factors were not taken into consideration. Moreover, it is also impossible to perform planning incorporating regulations depending on different situations, such as maintaining driving speed and suppressing posture change.

Ishigami *et al.* [20] proposed a method to generate multiple candidate path solutions, in order to perform the safe motion generation of the planetary rover. An appropriate path was selected from the solution candidates, based on evaluation criteria. They used the Dijkstra algorithm for path search to generate multiple candidates. In this method, when executing the path search, three types of evaluation criteria, such as degree of unevenness, degree of inclination, and path length information, were calculated from the environment map, and multiple paths were generated by changing the weight parameters of the evaluation criteria. Finally, a navigation simulation, based on the physical model of the vehicle, was conducted for each of the candidate path solutions to select an optimal path solution, with minimum energy consumption. This method achieved path planning, which considers changes in height direction and posture of the vehicle. However, the target environment in the proposal by Ishigami *et al.* was a relatively small area of 8 m^2, and the calculation time required to output the path solution within this area was approximately 47 min. Note that the computational complexity of the Dijkstra algorithm is defined as $\mathcal{O}(n^2)$, with respect to the number of nodes, and closely depends on the size of the environment; therefore, reliable path planning cannot be executed with feasible processing time when targeting a relatively larger scale environment map, and provided that the calculation time required for path solution search increases with the order of squares, as the number of nodes increases.

III. PROBLEM DESCRIPTION

As mentioned in the introduction, realizing the safe and reliable ability of autonomously traveling to a destination is one of the most important requirements for an off-road UGV in rough terrain environments. Hence, estimating the traversability and appropriate motion planning on rough terrain are very important tasks to meet these requirements. In this respect, we aim to propose a novel motion planning methodology for UGVs to navigate safely to a destination within convoluted environments, including rough terrain. By summarizing the previously conducted studies for vehicular motion planning, which were described in the previous section, we determined that the

following issues should be solved in this study. Therefore, when designing a novel motion planner, we need to consider the following.

1) All DoFs, namely the 6-DoFs of vehicular pose (position and orientation), which include height direction and roll and pitch angles [see Fig. 3(a)].
2) The unique characteristics of each vehicle, such as the size of the vehicle, minimum turning radius, or travelable maximum inclination angle, depending on the driving speed (details are given in Section IV-A).
3) Regulations necessary for vehicular operation depending on different situations, such as maintaining the driving speed and suppressing the change of posture (details are given in Section IV-B).
4) Feasible processing time to identify a solution, even in relatively large-scale environments.

As described previously, no previous study has reported solutions for all four issues. Hence, the purpose of this study is to establish a novel motion planner for off-road UGVs, which addresses all the aforementioned issues. Specifically, when the user specifies the initial pose and the target pose of the UGV with respect to the environmental map composed of a 3-D point cloud provided a priori, the motion planner should solve the problem of generating a path that connects these two states offline. In regard to the scale of the environment map, we have treated the scale spanning several hundred meters as large-scale in this study, and the proposed motion planner was applied to environmental maps with this size. Of course, the motion planner must work on a common environment that includes well-organized roads as well as an off-road environment, thus taking into account the vehicle characteristics and the predefined regulations.

IV. VEHICLE CHARACTERISTICS AND REGULATIONS

A. Vehicle Characteristics

Values for vehicle characteristics to be set depending on various elements, such as driving scheme, or mechanism design, must be considered as essential. The vehicle characteristics, for the off-road UGV, can be divided into two categories: characteristics related to the geometric shape of the vehicle and characteristics related to the operational restrictions of the vehicle. The former are values defining the overall size of the vehicle, such as the overall length, width, and height. On the other hand, the latter includes minimum turning radius or maximum inclination angle, which are set by the vehicle designer or the user. First, because we need to determine the path that the vehicle can actually follow, the minimum turning radius should be considered to generate a feasible vehicle motion. Next, regarding the maximum inclination angle, as the driving speed of the vehicle increases, the maximum inclination angle, with respect to the roll and pitch direction, decreases in general. On the other hand, when the vehicle is driving at low speed, the maximum inclination angle becomes large. This means that it is insufficient for simply generating a geometric path including only position data; thus, when the vehicle navigates the generated path, the driving speed and posture information must be considered concurrently, in order to prevent a rollover accident. In other words, when executing the motion planner, it is necessary to check
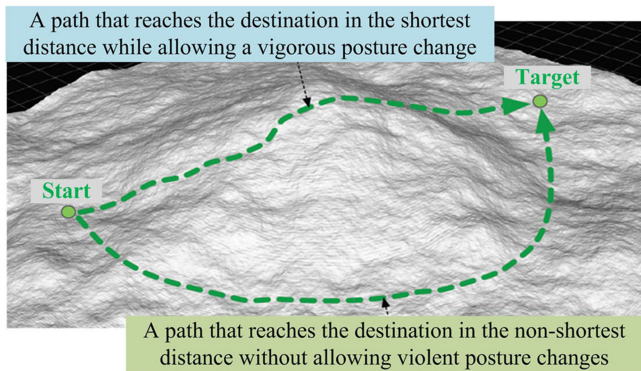
Fig. 1.   Conceptual image of motion planning based on different regulations on rough terrain.
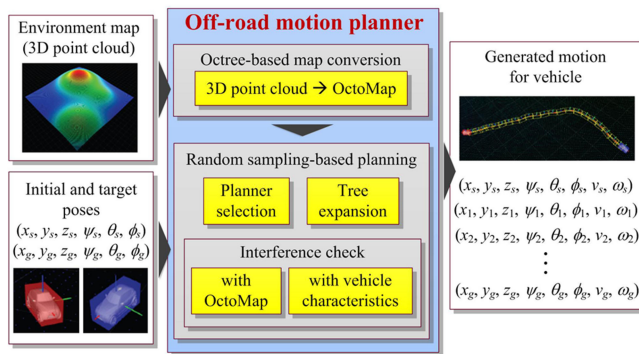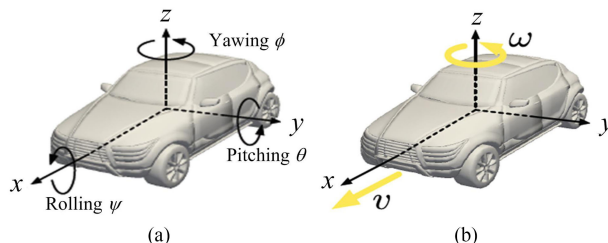

Fig. 2.   Overview of proposed motion planner.

Fig. 3.   Definition of local coordinate frame and states related to vehicle. (a) Orientation state (ψ, θ, ϕ). (b) Linear and angular control velocities (v, ω).

whether the vehicle's posture at each point on the path exceeds the maximum inclination angle or not. To summarize the discussion on vehicle characteristics, the off-road UGV traveling on rough terrain must carry out motion planning by considering vehicle characteristics, to prevent accidents and navigate safely.

B. Regulations for Vehicle Operation

Additionally, when solving motion planning problems for off-road UGVs traveling on rough terrain, we should consider the various requirements, depending on the target environment and purpose of UGV operation. This implies that the autonomous driving system should be operated based on several regulations in addition to producing a solution path that satisfies the vehicle characteristics described in the previous section. For example, in some cases, it is necessary to generate the vehicle's motion to the destination via the shortest route while allowing for vigorous posture change. On the other hand, some cases must generate motion without allowing a vigorous posture change instead of giving up the shortest route, from the viewpoint of ensuring safety during navigation. Fig. 1 is a conceptual image that shows the motion planner based on the different set regulations mentioned before, in an environment with rough terrain. The operational regulations for the off-road UGV include not merely a requirement regarding posture change and shortest route, but also the requirement of maintaining driving speed as much as possible while reducing speed fluctuations. In this latter case, by considering the vehicle characteristics described in the previous section, the output path should be generated under conditions that do not exceed the maximum inclination angle corresponding to the set desired driving speed. As described previously, to operate an off-road UGV under various situations, it is necessary to execute, in advance, a motion planner based on the set of regulations specified by the user.

V. Overview

The overall process of the proposed motion planner is illustrated in Fig. 2. The initial and target poses of the vehicle are the input data to the proposed motion planner, under the condition that an environment map consisting of a 3-D point cloud is pregiven. If solutions exist, the planner outputs one of the solutions satisfying the prespecified conditions (i.e., vehicle characteristics, regulations, etc.). The output data are a list of vehicular poses that connect the initial and target poses. However, of course, if neither the initial nor the target poses are valid (e.g., in the case of a collision with an obstacle), the planner will not work because the solution cannot be searched. Fig. 3(a) shows the local coordinate system of the vehicle adopted in this study. In addition, our motion planner can produce an output list including vehicle control input data as well as pose data. Here, (v, ω) denotes the control input data [i.e., control velocity, as shown in Fig. 3(b)] to the vehicle at state $(x, y, z, \psi, \theta, \phi)$. In this study, since the allowable inclination angle and turning radius can vary with the vehicular velocity (v, ω), as mentioned in the previous section, the control velocity affects the validity check of the specific pose on the terrain, based on the maximum inclination angle and maximum turning radius.

The octree-based map conversion process involves reducing the amount of 3-D point cloud data. If we use raw data in the 3-D point cloud, the computational burden would be very high, which makes the problem of global motion planning, in a relatively large-scale environment, very difficult. Hence, it is necessary to reduce the amount of data by using an efficient data structure for the 3-D environment model. Several ways to represent a 3-D environment model have been developed [21]–[24]. Among them is the OctoMap library, which has been used widely to manage a dense 3-D environment model, and has been utilized to convert a 3-D point cloud to address this problem [24]. The OctoMap data structure consists of irregular voxels, which is based on an octree structure that is very efficient for memory management, in comparison to a point cloud-based structure. One of the great advantages of using the OctoMap is that the interference checking problem between the vehicle and the 3-D environment model becomes a tree search problem, which has the advantage of efficient search time with

low time complexity. Therefore, when converting to OctoMap format from 3-D point cloud data, the processing time required for the solution search can be drastically reduced; thus, we can solve the motion planning problem since the processing time becomes feasible, even in a relatively large-scale environment including large amounts of data.

During the random sampling based planning process, a 6-D configuration space, composed of 3-D position (x, y, z) and 3-D orientation (ψ, θ, ϕ) states, is generated. Then, motion planning is executed to find a solution, which connects from the initial configuration to the target configuration. Here, the random sampling based planner is a widely used method that can efficiently search a high-dimensional configuration space. We can select the appropriate algorithm among the various random sampling based planners (e.g., RRT, PRM, PDST, etc.). When expanding the tree structure in the random sampling based algorithm, we should consider the motion model of a vehicle and 3-D terrain information, which can be calculated from OctoMap, in order to check for interference with configuration obstacles. The interference check module determines whether the configuration of a generated random node is included in the configuration obstacles or not. When checking the state of the random node, we consider not only physical interference between a vehicle and OctoMap, but also vehicle characteristics, so that it is possible to plan a motion that satisfies vehicle characteristics and user-defined regulations. Furthermore, since the random sampling based algorithm already take the kinematic model and the characteristics for the actual vehicle into consideration, our proposed motion planning scheme does not need any postprocessing, such as smoothing algorithms to manipulate the path solution so that the actual vehicle can follow.

VI. RANDOM SAMPLING BASED MOTION PLANNING

To realize realistic motion planning that considers the 6-DoF state of off-road UGV, we introduce the concept of configuration space in the proposed method. The configuration space consists of six dimensions including three position variables and three orientation variables. Motion planning is executed to find a path from the initial configuration state to the target configuration state, within the entire configuration space. A planning method in the configuration space can be classified into two categories: the planning scheme in discretized target space and the planning scheme in continuous target space. Among them, the Dijkstra algorithm or the A* search algorithm belong to the former, whereas the latter includes a potential field based method [25] or random sampling based method, such as RRT [9], PRM [10], PDST [11], etc. In the case of the former method, which discretizes the target space, the entire space for every possible state of the vehicle must be mapped to the configuration space completely; therefore, it requires a relatively long calculation time, in general. Especially, the problem related to calculation time becomes a fatal flaw in the case of this study, which targets relatively large-scale environments; thus, in this study, the former method cannot be applied to the motion planning problem that should deal with such environments. In this sense, we solved the motion planning problem based on a random sampling based scheme, which was classified as the latter method that treats the target space as continuous,
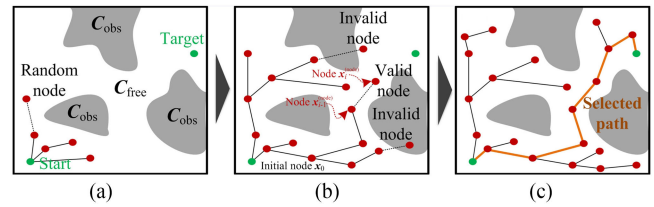

Fig. 4. Random sampling process from start to target configuration in configuration space.

since it does not require any preprocessing, and can perform high-speed search even in multidimensional space. Here, "random sampling" means randomly generating a state corresponding to a certain configuration within the entire configuration space.

This section briefly summarizes the common processing of various random sampling based methods. Each method has its own characteristics and advantages. Further details are provided in the references [9]–[11]. A conceptual image of random sampling in the configuration space is shown in Fig. 4. In this study, the configuration corresponds to a 6-DoF vehicle pose $(x, y, z, \psi, \theta, \phi)$, and the configuration obstacles $\mathbb{C}_{\mathrm{obs}}$ represent areas where unrealizable vehicle poses are mapped to the configuration space. The path search in the motion planning problem can be rephrased as a problem of finding a path solution, within $\mathbb{C}_{\mathrm{free}}$, from the initial configuration $\boldsymbol{x}_s = [x_s \, y_s \, z_s \, \psi_s \, \theta_s \, \phi_s]^\top (\in \mathbb{C}_{\mathrm{free}})$ to the target configuration $\boldsymbol{x}_g = [x_g y_g z_g \psi_g \theta_g \phi_g]^\top (\in \mathbb{C}_{\mathrm{free}})$. $\mathbb{C}_{\mathrm{free}}$ refers to the area set of possible configurations that the vehicle can take, and is determined by the shape of the vehicle and terrain in the workspace. In general, this is difficult to describe explicitly because their interaction in high-dimensional space is complicated. On the other hand, if a specific configuration is given, it is relatively easy to check whether it belongs to $\mathbb{C}_{\mathrm{free}}$ or $\mathbb{C}_{\mathrm{obs}}$; thus, by efficiently implementing the process of verification, high-speed motion planning can be realized even in high-dimensional space. In Section VIII, the verification process is described in detail. As shown in Fig. 4(b), the random sampling based motion planner expends random nodes that correspond to a configuration from the initial configuration to the target configuration, only within a $\mathbb{C}_{\mathrm{free}}$ space (i.e., an area that does not belong to a configuration obstacle), by using a method such as random number generation. Additionally, it concatenates them while checking for interference with the configuration obstacles $\mathbb{C}_{\mathrm{obs}}$. As a result, a path solution that does not interfere with the configuration obstacle $\mathbb{C}_{\mathrm{obs}}$ can be found within the expanded tree structure, as shown in Fig. 4(c). In the next section, the tree expansion scheme in the proposed random sampling based motion planner is described.

VII. TREE EXPANSION

The proposed method takes the motion model of a vehicle and terrain information from OctoMap into consideration while extending a new tree structure, from the tree of previous steps, by generating random nodes in configuration space. Fig. 5 is an overview of the tree expansion methodology of the proposed method. The process to generate a new node $\boldsymbol{x}_i^{(\mathrm{node})} = [x_i \, y_i \, z_i \, \psi_i \, \theta_i \, \phi_i]^\top$ from a previous node
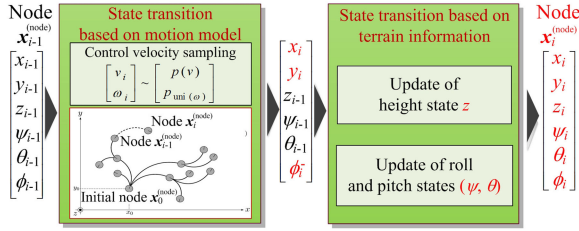
Fig. 5. Overview of tree expansion methodology that transits each random node.
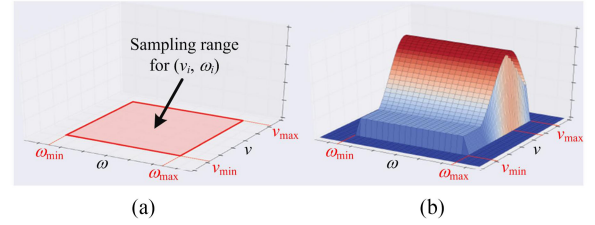

Fig. 6. Biased sampling scheme for control velocity generation. (a) Sampling range. (b) Final probability distribution for generating control velocity.

$\boldsymbol{x}_{i-1}^{(\text{node})} = [x_{i-1}\ y_{i-1}\ z_{i-1}\ \psi_{i-1}\ \theta_{i-1}\ \phi_{i-1}]^\top$ divided into two transitions, which are performed based on the motion model of the vehicle and terrain information (i.e., height and inclination information from OctoMap), respectively. In other words, the random sampling process is performed in a 3-DoF (x, y, ϕ) configuration space to apply the vehicles motion model given that the control of the vehicle navigation on the ground is generally carried out in 3-DoF (x, y, ϕ) space. The remaining state variables (z, ψ, θ) are then directly calculated and assigned using the terrain information because the OctoMap contains enough information about the terrains height and inclination. The details of each process expending the tree structure composed of random nodes are described ahead.

A. State Transition Based on Vehicle Motion Model

If random nodes are generated based on the moving ability of the vehicle, when expanding the tree structure for searching in the configuration space, a more realistic motion can be generated as the solution. Fig. 4(b) shows the expanding tree structure in the configuration space. Nodes are generated only at the location where the vehicle is able to reach, and these nodes are connected. In other words, considering the moving ability of the vehicle leads to the prevention of the generation of a random node to a location that is, realistically, impossible to reach.

To this end, random sampling is first performed to generate the control input velocity of the vehicle. The reason for sampling the control input velocity instead of directly sampling the node states is to incorporate the user-defined regulations described in Section IV-B into the planned results. If the velocities between the states are calculated after the states are sampled, we cannot guarantee whether or not the velocities will comply with the regulations. Herein, the appropriate probability distribution $p(v)$ and the upper and lower limits for each control velocity $(v_{\min}, v_{\max}, \omega_{\min}, \omega_{\max})$ are defined for random number generation to carry out the motion planning with consideration to the regulations. We can take a uniform distribution within the set range $(v_{\min}, v_{\max}, \omega_{\min}, \omega_{\max})$ as the simplest way to consider random number generation to generate control velocity (v_i, ω_i). However, if the control velocity (v_i, ω_i) is generated according to a distribution with no bias, such as uniform distribution, the dispersion of generated velocities would be very large, which leads to infeasible or unstable vehicle motion. In particular, we cannot satisfy one of the requirements for this study, specifically, the one that generates the motion satisfying the regulations, such as suppressing large speed changes (i.e., maintaining driving speed), if the linear control velocity

v_i of each node becomes a completely random value in this case.

To remedy this problem, we propose a biased sampling scheme, to incorporate such regulations when generating the linear control velocities v_i. We apply the probability distribution function $p(v)$ to the implementation of biased sampling for the linear control velocity v_i given to each node $\boldsymbol{x}_i^{(\text{node})}$ as follows:

$$p(v) = \begin{cases} f_1(v), & f_1(v) > f_2(v) \\ f_2(v), & \text{otherwise} \end{cases} \tag{1}$$

$$f_1(v) = \frac{1}{\sqrt{2\pi v_\sigma^2}} \exp\left\{ -\frac{(v - v_\mu)^2}{2v_\sigma^2} \right\} \tag{2}$$

$$f_2(v) = p_{\text{uni}(v)}. \tag{3}$$

Here, $f_1(v)$ and $f_2(v)$, respectively, follow v_μ mean distribution with v_σ^2 variance $\mathcal{N}(v_\mu, v_\sigma^2)$ and uniform distribution $p_{\text{uni}(v)}$. Therefore, in this study, biased sampling is performed based on the mixed probability distribution of Gaussian and uniform distribution, as indicated in (1)–(3). Additionally, the lower limit velocity $v_{\min}$ and upper limit velocity $v_{\max}$ are set based on the performance of the vehicle to be operated. The probability becomes 0, when the range is $v < v_{\min}$ or $v > v_{\max}$, so that it is possible to generate a control velocity that does not go beyond the performance of the vehicle. With respect to angular velocity sampling ω_i, we apply a uniform probability distribution $p_{\text{uni}(\omega)}$. The upper limit velocity $v_{\max}$, and lower limit velocity $v_{\min}$ are also set based on the performance of the vehicle in the same manner as in the case of linear velocity. Fig. 6(a) shows how the sampling range $(v_{\min}, v_{\max}, \omega_{\min}, \omega_{\max})$ is set for linear and angular velocity, respectively. Random sampling is performed within this range in order to assign the control velocity (v_i, ω_i) to the node $\boldsymbol{x}_i^{(\text{node})}$. Fig. 6(b) shows the visualization of the final probability distribution, whose range [see Fig. 6(a)] is set in (1).

The advantages for applying biased sampling using the probability distribution represented in Fig. 6(b), are as follows. In this approach, the control velocities close to the mean value v_μ are most frequently generated and assigned to random nodes; therefore, by adjusting v_μ and v_σ, which are parameters of the Gaussian distribution (2), it is possible to generate appropriate motion, which reflects the regulations described in Section III-B, such as the maintenance of specific speed and variation of its fluctuation, respectively. On the other hand, if all control velocities are sampled around v_μ, a solution may not be determined owing to the terrain conditions. Even in such

situations, the proposed approach can find a solution that is able to attain the goal because the control values, which deviate from v_μ for a short duration, can also be generated with low probability by combining uniform distribution $p_{\text{uni}(v)}$; therefore, it is possible to cope more flexibly with respect to the solution search according to various terrain conditions. The user can define the allowable deviation of the sampling from v_μ by manipulating $p_{\text{uni}(v)}$ of the uniform distribution (3). Consequently, the state of the new node $\boldsymbol{x}_i^{(\text{node})}$ is generated from node $\boldsymbol{x}_{i-1}^{(\text{node})}$ by the motion model based on the control velocities (i.e., linear velocity v_i and angular velocity ω_i), which are generated by the random sampling process described before. Therefore, in this transition step, the configuration of each node is transited based on the motion model of the vehicle, which is defined for a state of position (x, y), and the yaw angle ϕ as follows:

$$x_i = x_{i-1} - \frac{v_i}{\omega_i} \sin \phi_{i-1} + \frac{v_i}{\omega_i} \sin (\phi_{i-1} + \omega_i \Delta t) \quad (4)$$

$$y_i = y_{i-1} + \frac{v_i}{\omega_i} \cos \phi_{i-1} - \frac{v_i}{\omega_i} \cos (\phi_{i-1} + \omega_i \Delta t) \quad (5)$$

$$\phi_i^- = \psi_{i-1} + \omega_i \Delta t \quad (6)$$

where i is the index of the node and Δt is the time period of the control input. v_i and ω_i denote the control input for the linear and angular velocities, which are, respectively, generated from the biased sampling described previously. A new node $\boldsymbol{x}_i^{(\text{node})}$ is additionally generated from the node $\boldsymbol{x}_{i-1}^{(\text{node})}$ based on (4)–(6) and these are newly added to the tree structure. Here, superscript "$-$" in (6) indicates the yaw angle state before it is updated by terrain information. Note that the motion model that uses the linear and angular velocities (v, ω) as the control input data is the most generalized model. It is easily applicable to various types of vehicles (e.g., Ackerman steering, differential steering, etc.), given that we only need to convert the control input data from (v, ω) to other inputs that correspond to the vehicle types in the instances where vehicular control is applied to allow vehicles to follow the generated paths.

B. State Update Based on Terrain Information of OctoMap

After transition based on the motion model, the configuration of the transited node $\boldsymbol{x}_i^{(\text{node})}$ is updated for height z and roll and pitch angles (ψ, θ) based on terrain information. First, this transition step utilizes the height information from OctoMap to update the height state of the new node $\boldsymbol{x}_i^{(\text{node})}$. The detailed procedure for this is as follows. Some voxels constituting the OctoMap in the vicinity of the newly generated node position (x_i, y_i), at the previous transition step are selected. Here, as shown in Fig. 7(a), the definition of the vehicles vicinity area is determined as a rectangular area around the vehicle, based on its size. Therefore, the voxels that cover a rectangular area around the nodal position (x_i, y_i) are selected. Then, the height information z_i of the new node $\boldsymbol{x}_i^{(\text{node})}$ is updated according to the following equation, which retrieves the maximum height information among the selected voxels

$$z_i = z_{\max}^{(\text{vox})} = \max\{z_k^{(\text{vox})} \mid k = 1, 2, \ldots, N^{(\text{vox})}\} \quad (7)$$
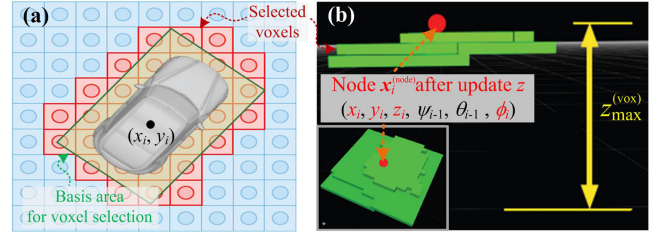

Fig. 7. Process used to update height state based on terrain information from OctoMap. (a) Voxel selection belonging to rectangular region. (b) Updated height state.
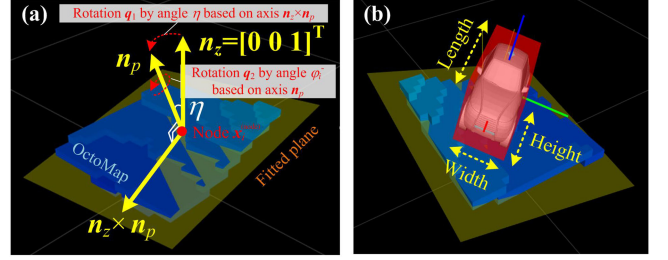

Fig. 8. Process to update orientation state based on terrain information from OctoMap. (a) Rotation phases based on fitted plane. (b) Updated orientation state depicted with vehicle model.

where k and $N^{(\text{vox})}$ denote an index for distinguishing each voxel belonging to the rectangular region and their total number. $z_k^{(\text{vox})}$ represents the height information of each voxel. Fig. 7(b) shows how height state z_i of node $\boldsymbol{x}_i^{(\text{node})}$ is assigned according to (7) that retrieves the maximum height information among the selected voxels.

Next, the orientation state of the new node $\boldsymbol{x}_i^{(\text{node})}$ is also updated based on the inclination information from OctoMap. Fig. 8 shows the overall process to update the orientation state. As shown in Fig. 8(a), this update process selects voxels belonging to a rectangular area in the same manner as update step of the height information, to perform PCA, on the center point of each voxel, as follows:

$$\bar{\boldsymbol{x}}^{(\text{vox})} = \frac{1}{N^{(\text{vox})}} \sum_{k=1}^{N^{(\text{vox})}} \boldsymbol{x}_k^{(\text{vox})} \quad (8)$$

$$\boldsymbol{C}^{(\text{vox})} = \frac{1}{N^{(\text{vox})}} \sum_{k=1}^{N^{(\text{vox})}} (\bar{\boldsymbol{x}}_k^{(\text{vox})} - \boldsymbol{x}_k^{(\text{vox})})(\bar{\boldsymbol{x}}_k^{(\text{vox})} - \boldsymbol{x}_k^{(\text{vox})})^\top \quad (9)$$

where $\boldsymbol{x}_k^{(\text{vox})} = [x_k^{(\text{vox})} \ y_k^{(\text{vox})} \ z_k^{(\text{vox})}]^\top$ denotes the coordinates of a center point for each voxel. By performing eigenvalue analysis on the covariance matrix $\boldsymbol{C}^{(\text{vox})}$, eigenvalues $\lambda_1^{(\text{vox})} \geq \lambda_2^{(\text{vox})} \geq \lambda_3^{(\text{vox})}$ and eigenvectors $\boldsymbol{e}_1^{(\text{vox})}, \boldsymbol{e}_2^{(\text{vox})}$, and $\boldsymbol{e}_3^{(\text{vox})}$ can be calculated, respectively. Here, $\boldsymbol{e}_1^{(\text{vox})}$ indicates the direction that has maximum variance with respect to the position of each voxel and $\boldsymbol{e}_j^{(\text{vox})}$ is orthogonal to $\boldsymbol{e}_m^{(\text{vox})}$ $(m = 1, \ldots, j-1)$. Therefore, we can fit the plane for the terrain, which has a normal vector $\boldsymbol{e}_3^{(\text{vox})}$, as shown in Fig. 8(a). According to this approach, the orientation state $(\psi_i, \theta_i, \phi_i)$ of node $\boldsymbol{x}_i^{(\text{node})}$ is calculated using the rotation matrix based on this fitted plane. In the

first phase, the cross product $\boldsymbol{n}_z \times \boldsymbol{n}_p$ and the angle η between the vertical unit vector $\boldsymbol{n}_z = [0\,0\,1]^\top$ and the normal vector of the approximate plane $\boldsymbol{n}_p = \boldsymbol{e}_3^{(\mathrm{vox})} = [e_{3x}^{(\mathrm{vox})}\ e_{3y}^{(\mathrm{vox})}\ e_{3z}^{(\mathrm{vox})}]^\top$, are, respectively, calculated as follows:

$$\boldsymbol{n}_z \times \boldsymbol{n}_p = \left[-e_{3y}^{(\mathrm{vox})}\ e_{3x}^{(\mathrm{vox})}\ 0 \right]^\top \tag{10}$$

$$\eta = \arccos \frac{\boldsymbol{n}_z \cdot \boldsymbol{n}_p}{\|\boldsymbol{n}_z\|\|\boldsymbol{n}_p\|}. \tag{11}$$

In the second phase, two rotations are performed continuously as indicated in Fig. 8(a): rotation by angle η based on the rotation axis $\boldsymbol{n}_z \times \boldsymbol{n}_p$, and rotation by angle ϕ_i^- based on rotation axis $\boldsymbol{n}_p$. Here, ϕ_i^- is the yaw angle, which was calculated by the motion model expressed by (6). This rotation process can be expressed as the product of two quaternion expressions $\boldsymbol{q}_2\boldsymbol{q}_1$, as follows:

$$\boldsymbol{q}_2\boldsymbol{q}_1 = \left[q_x\ q_y\ q_z\ q_w \right]^\top \tag{12}$$

$$\boldsymbol{q}_1 = \left[-e_{3y}^{(\mathrm{vox})} \sin\frac{\eta}{2}\ e_{3x}^{(\mathrm{vox})} \sin\frac{\eta}{2}\ 0\ \cos\frac{\eta}{2} \right]^\top \tag{13}$$

$$\boldsymbol{q}_2 = \left[e_{3x}^{(\mathrm{vox})} \sin\frac{\phi_i^-}{2}\ e_{3y}^{(\mathrm{vox})} \sin\frac{\phi_i^-}{2}\ e_{3z}^{(\mathrm{vox})} \sin\frac{\phi_i^-}{2}\ \cos\frac{\phi_i^-}{2} \right]^\top. \tag{14}$$

Finally, to update the orientation state $(\psi_i, \theta_i, \phi_i)$ for the random node $\boldsymbol{x}_i^{(\mathrm{node})}$, we transform the quaternion notation to roll-pitch-yaw notation as follows:

$$\psi_i = \arctan \frac{2(q_w q_x + q_y q_z)}{1 - 2(q_x^2 + q_y^2)} \tag{15}$$

$$\theta_i = \arcsin \{2(q_w q_y - q_z q_x)\} \tag{16}$$

$$\phi_i = \arctan \frac{2(q_w q_z + q_x q_y)}{1 - 2(q_y^2 + q_z^2)}. \tag{17}$$

Through the steps presented before, the orientation state $(\psi_i, \theta_i, \phi_i)$ of the new node $\boldsymbol{x}_i^{(\mathrm{node})}$ is completely updated based on the terrain information from the OctoMap, as shown in Fig. 8(b).

As a result, the vehicular model that corresponds to each node closely interfaces the terrain. Note that the processing steps introduced in this section to achieve state updates represent an approximate processing methodology given that the modeling of the vehicle as a rigid body—represented as a rectangular parallelepiped—is not guaranteed to tangentially interface a nonflat terrain completely. In general, because the vehicle is in contact with the ground via its rubber tires, the updated states are thus the most stable poses in contact with the terrain. Therefore, this approximation process is valid.

VIII. DESIGN OF CONFIGURATION OBSTACLES

This section describes the details of a design method for configuration obstacles defined in configuration space. While generating random nodes to expend tree structure in the configuration space, the validity of the nodes, concerning whether their corresponding configurations are included in the area of the configuration obstacles $\mathbb{C}_{\mathrm{obs}}$, must be verified. The only nodes whose validity is guaranteed with respect to the configuration of the vehicle are connected to each other to expand the tree structure. Thus, in this study, the design of the evaluation criteria, for verifying the validity of the node, is equivalent to the design
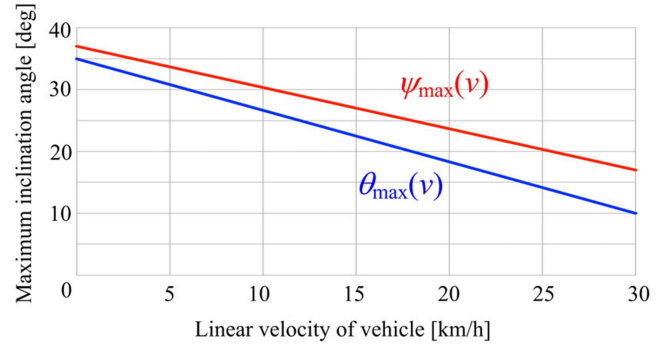

Fig. 9. Example of relationship between linear velocity v and corresponding maximum inclination angles $(\psi_{\max}, \theta_{\max})$.

of the configuration obstacle. From the point of view of avoiding physical obstacles and considering vehicle characteristics, we designed two evaluation criteria for the design of configuration obstacles: interference check with OctoMap, and vehicle characteristics check. Both criteria are presented in detail in the following sections.

A. Interference Check With OctoMap

First, the evaluation criteria check whether the vehicle itself, assumed to exist on the node, interferes with OctoMap based on the pose of each random node $\boldsymbol{x}_i^{(\mathrm{node})} = [x_i\ y_i\ z_i\ \psi_i\ \theta_i\ \phi_i]^\top$ generated in the configuration space. As shown in Fig. 8(b), we define a 3-D model representing the geometry of the vehicle by using the overall size parameters (i.e., length, width, and height), mentioned in Section III-A. The interference check between the 3-D model of the vehicle, corresponding to the random node, and OctoMap, can be written as follows:

$$\forall (x^{(\mathrm{vm})}, y^{(\mathrm{vm})}, z^{(\mathrm{vm})}) \notin \mathbb{M}_{\mathrm{oct}} \ \rightarrow\ \boldsymbol{x}_i^{(\mathrm{node})} \in \mathbb{C}_{\mathrm{free}} \tag{18}$$

$$\exists (x^{(\mathrm{vm})}, y^{(\mathrm{vm})}, z^{(\mathrm{vm})}) \in \mathbb{M}_{\mathrm{oct}} \ \rightarrow\ \boldsymbol{x}_i^{(\mathrm{node})} \in \mathbb{C}_{\mathrm{obs}} \tag{19}$$

where $(x^{(\mathrm{vm})}, y^{(\mathrm{vm})}, z^{(\mathrm{vm})})$ denotes an arbitrary 3-D position belonging to the 3-D model of the vehicle corresponding to node $\boldsymbol{x}_i^{(\mathrm{node})}$. $\mathbb{M}_{\mathrm{oct}}$ represents a set of voxels with regular hexahedron structure constituting the OctoMap. In this study, we utilized the flexible collision library [26] in order to implement the inference check between the 3-D model of the vehicle and the set of voxels constituting the OctoMap.

B. Vehicle Characteristic Check

The second evaluation criteria check whether the generated random node satisfies the condition defined as the vehicle characteristics by considering the 6-DoF state of the node $(x, y, z, \psi, \theta, \phi)$ and control velocity (v, ω). This check process is performed based on two criteria: maximum inclination angle and minimum turning radius.

1) Maximum Inclination Angle: A typical example of the relationship between the linear velocity of the vehicle v and corresponding maximum inclination angles $(\psi_{\max}, \theta_{\max})$ is represented in Fig. 9. The relationship between these two is modeled as a linear monotone decreasing function, in this case, provided that the maximum allowable inclination angles, corresponding to both the roll and pitch angles, become smaller as the vehi-

Fig. 10. Suzuki Jimny customized for fully autonomous driving on rough terrain, which is assumed as off-road UGV in experiments.

cle drives at higher speed. Note that this relationship should be set differently depending on various situations, such as vehicle performance or user intention. These evaluation criteria can be written as follows:

$$|\psi_i| \leq \psi_{\max}(v_i) \text{ and } |\theta_i| \leq \theta_{\max}(v_i) \rightarrow \boldsymbol{x}_i^{(\text{node})} \in \mathbb{C}_{\text{free}} \quad (20)$$

$$|\psi_i| > \psi_{\max}(v_i) \text{ or } |\theta_i| > \theta_{\max}(v_i) \rightarrow \boldsymbol{x}_i^{(\text{node})} \in \mathbb{C}_{\text{obs}} \quad (21)$$

where ψ_i and θ_i denote the roll and pitch angles of node $\boldsymbol{x}_i^{(\text{node})}$, which are calculated, respectively, by the methodology described in Section VI-B. v_i is the linear velocity corresponding to node $\boldsymbol{x}_i^{(\text{node})}$. $\psi_{\max}(v_i)$ and $\theta_{\max}(v_i)$ represent the maximum allowable inclination angles obtained by the graph shown in Fig 9.

2) Maximum Turning Radius: Finally, minimum turning radius should also be considered in the design, to evaluate whether node $\boldsymbol{x}_i^{(\text{node})}$ belongs to a configuration obstacle. The turning radius of the vehicle is defined as $|v/\omega|$. Therefore, the evaluation criteria for the minimum turning radius can be defined as follows:

$$|v_i/\omega_i| \leq r_{\min} \rightarrow \boldsymbol{x}_i^{(\text{node})} \in \mathbb{C}_{\text{free}} \quad (22)$$

$$|v_i/\omega_i| > r_{\min} \rightarrow \boldsymbol{x}_i^{(\text{node})} \in \mathbb{C}_{\text{obs}} \quad (23)$$

where $|v_i/\omega_i|$ denotes the turning radius that corresponds to the node $\boldsymbol{x}_i^{(\text{node})}$. $r_{\min}$ denotes the minimum turning radius, which should be defined by the user in advance. It should be set in consideration to various conditions, such as the performance of the vehicle.

Consequently, only random nodes satisfying all conditions of (18), (20), and (22) are added to the new tree structure. On the other hand, nodes satisfying only one of the (19), (21), and (23) are removed.

IX. EXPERIMENTAL RESULTS

In the experiments, Suzuki Jimny, which is widely known for very high off-road performance, was assumed as the off-road UGV. Fig. 10 shows a Suzuki Jimny customized for fully autonomous driving on rough terrain. We specify four fixed parameters in advance, for all experiments, with reference to vehicle characteristics: vehicle length 3.4 m, weight 1.5 m, height 1.7 m, and minimum turning radius 4.8 m. In addition, the function model for the relationship between linear velocity and maximum inclination angles is as follows:

$$\psi_{\max}(v) = -0.68v + 37 \quad (24)$$

$$\theta_{\max}(v) = -0.83v + 35. \quad (25)$$

Here, the units of angles $(\psi_{\max}, \theta_{\max})$ and linear velocity v are deg and km/h, respectively. The graphs for (24) and (25) are represented in Fig. 9. We used (24) and (25) to determine the validity of the generated random nodes based on the maximum inclination angles by considering the vehicular speed. In other words, these equations are used to calculate $\psi_{\max}$ and $\theta_{\max}$ in the evaluation criteria of (20) and (21). Note that the coefficients of these monotonically decreasing functions were set in consideration of the performance and safety of the UGV shown in Fig. 10. In contrast, the following parameters are set differently for each experiment by considering the regulations or user intention, namely time period of control input Δt, upper and lower limits for linear velocity $(v_{\min}, v_{\max})$, and Gaussian distribution parameters (v_μ, v_σ) for random number generation. A laptop with an Intel Core i7-3612QM CPU and 12.0-GB memory was used to execute the proposed motion planner.

In our implementation of random sampling, we set the sampling period Δt (i.e., the control period of the vehicle) in (4)–(6) to 0.5 s. Hence, the extension length of each node depends on the control velocity v sampled by the probability density function (1). In addition, apart from the processes used to generate the random nodes, as described in Section VII-A, a goal bias node and corresponding control velocity are generated with a probability of 5 % on the direction toward the target point. As a result, it is possible to implement a planner with a bias that is directed to the destination. Of course, the validity check on all generated nodes, including the goal biased node, is performed in accordance to the evaluation criteria described in Section VIII. Thus, only valid nodes remain.

A. Simulation Environment

In the simulation experiments, we generated the 3-D point cloud for an artificial off-road environment, as shown in Fig. 11, to verify the effectiveness of the proposed off-road motion planner. The size of the entire information map was approximately 200×200 m^2 and its point cloud contained millions of points. Thus, it could be considered a relatively large-scale environment compared to environments covered by previous studies. Moreover, the environment contained a high mountain at the middle of the environment, as well as a large amount of rough terrain; therefore, the planner was required to decide whether to avoid or to directly pass through the mountain, depending on the performance of the vehicle or predefined regulations. Here, we conducted experiments of two speed regulation patterns on the same environment: velocity sampling based on 10 km/h (pattern 1) and velocity sampling based on 30 km/h (pattern 2). The intended regulation in the experiment of pattern 1 was that the planner should allow a vigorous posture (i.e., roll and pitch angles) change when passing through rough terrain, since low linear velocities of approximately 10 km/h were sampled. In stark contrast with pattern 1, the motion planner of pattern 2 should suppress violent posture fluctuations; thus, it should generate a motion that avoids steep terrain because high linear velocities of approximately 30 km/h were randomly generated.
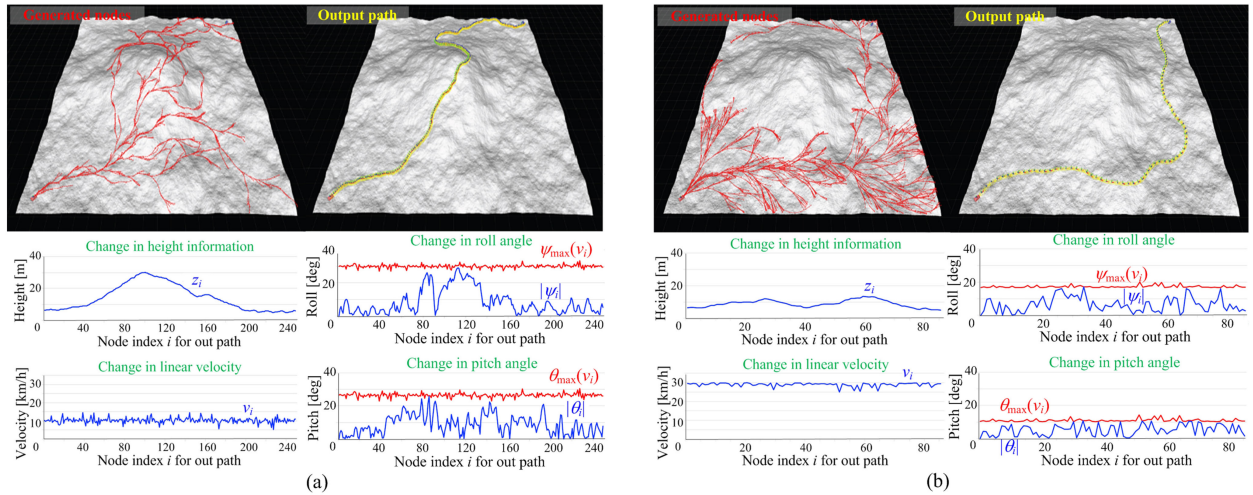
Fig. 11. Experimental results in simulation environment. (a) All generated nodes, output path as solution, and changes of several variables for each node on generated motion in case of pattern 1. (b) All generated nodes, output path as solution, and changes of several variables for each node on generated motion in case of pattern 2.

TABLE I
PARAMETERS FOR EACH EXPERIMENTAL CONDITION

Parameter	Pattern 1	Pattern 2
Time period of input Δt	0.5 s	0.5 s
Lower limit v_{min}	5.0 km/h	25.0 km/h
Upper limit v_{max}	15.0 km/h	35.0 km/h
Desired velocity v_μ	10.0 km/h	30.0 km/h
Standard deviation v_σ	1.0 km/h	1.0 km/h

Fig. 12. Repeated experimental results using each parameter set for pattern 1 (red paths) and pattern 2 (blue paths).

Table I summarizes a set of variable parameters in view of these regulations, for each experimental condition. In the simulation environment experiments, the initial and target poses were set at the bottom-left side and upper-right side of the environment, respectively, to generate motion for as wide a region as possible. The PDST [11] planner was chosen as the motion planner. The experimental results for patterns 1 and 2 are shown in Fig. 11. Here, the time required to find solutions in both cases was within 5 s, which is a feasible processing time, and the total number of generated nodes constituting the tree were 7873 and 6259, respectively. Among them, the number of nodes on the output paths were 251 and 87, respectively. Their lengths were 366.0 and 346.3 m respectively.

Based on the intended regulations mentioned before, in the case of the pattern 1, many random nodes were generated in a region with an undulating mountain in the central part of the map; thus, a solution path was generated through the undulating environment, as shown in Fig. 11(a), given that the allowable inclination angles in the roll and pitch directions, at linear velocities of approximately 10 km/h are relatively large, as represented in Fig. 9, (24), and (25). On the other hand, in contrast to pattern 1, the motion planner with the parameters of pattern 2 could not generate random nodes on the central part of the map, therefore, it made a detour to avoid the undulating mountain, as shown in Fig. 11(b), because the allowable inclination angles in the roll and pitch directions, at linear velocities of approximately 30 km/h, were relatively small. Graphs in Fig. 11 illustrate the changes to each node on the generated
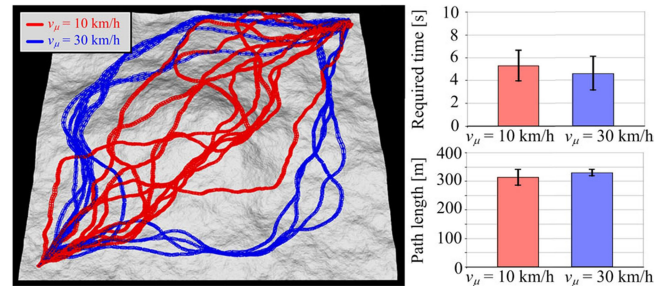
motion, namely changes in the height direction, linear velocity, roll angle, and pitch angle. Note that all nodes did not exceed the permissible inclination angles $\psi_{max}(v)$ and $\theta_{max}(v)$ calculated from (24) and (25).

Fig. 12 shows repeated experimental results based on each parameter for patterns 1 and 2. We replicated each experiment 10 times. Most of the results of pattern 1 passed through the undulating mountain in the center part. On the other hand, all the results of pattern 2 circumvented that area. The times required to produce solutions ranged between 3–6 s approximately. Furthermore, the standard deviations associated with the required times as well as path lengths for the repeated experimental results were not very large. Additionally, we have successfully conducted additional experiments in a much larger off-road environment (1000×1000 m^2) under conditions that require fairly long paths as solutions. The time taken to identify the solutions was all within 200 s, which is a relatively feasible processing time. Of course, the computation time depends on the size of the environment. However, in contrast to the Dijkstra algorithm described in Section II, our approach does not lead to computational time increases as a function of the square of the number of nodes. Consequently, the aforementioned results indicate that we could design a novel motion planner, which could generate motion within feasible processing times by considering all the
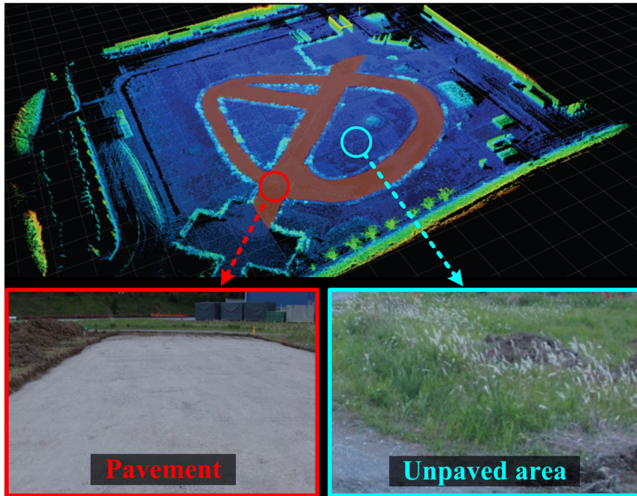
Fig. 13. Actual experimental environment including both paved and unpaved area.

TABLE II
SPECIFICATIONS OF VELODYNE LiDAR SENSOR HDL-32E

Specification item	Value	Unit
Laser class	1	-
Frame rate	5–20	Hz
Horizontal field-of-view	360	deg
Vertical field-of-view	+10 to -30	deg
Number of channels	32	-
Measurement range	1 to 70	m
Accuracy	Less than 2	cm
Output	Up to 800,000	points/s

DoFs, vehicle characteristics, and predefined regulations, in a relatively large, off-road environment.

B. Real Environment

To verify its applicability in a typical organized outdoor environment, we conducted additional experiments in a real environment, which had both unpaved area and paved road, as shown in Fig. 13. We built the information map (i.e., 3-D point cloud data) of the environment by using a mobile mapping system, as shown in Fig. 10, equipped with a global positioning system and a Velodyne laser imaging detection and ranging (LiDAR) HDL-32E sensor in advance. The LiDAR sensor was installed at a height of 2.4 m on the z-axis of the vehicle's coordinate frame. The specifications of the HDL-32E are shown in Table II. The entire size of the built map information was 208×184 m^2 and its point cloud contained 4,615,845 points. We performed a substantial number of experiments under various conditions, such as changing initial and target poses, vehicle parameters, regulations, and planner selection. Among them, Figs. 14 and 15 show experimental results at different initial and target poses, which, respectively, apply the RRT [9] and PDST [11] planners to the pattern 1 parameters listed in Table I. The details of the results were as follows. In regard to the RRT planner [see Fig. 14], the times required to identify the solutions were approximately 1 and 3 s, respectively. The total number of nodes that represented the entire set of trees were, respectively, 707 and 1980. The total
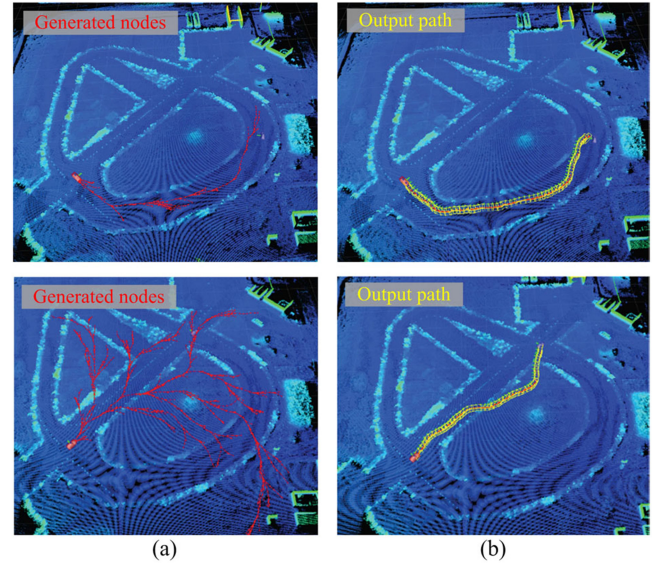

(a)   (b)

Fig. 14. Experimental results which apply the RRT [9] in a real environment. (a) Generated nodes. (b) Output path presented as solutions.
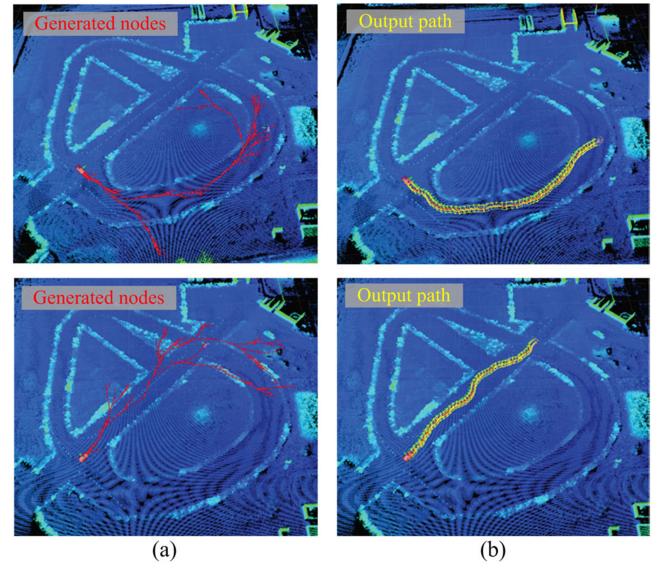

(a)   (b)

Fig. 15. Experimental results which apply the PDST [11] in real environment. (a) Generated nodes. (b) Output path presented as solutions.

number of output path solutions were, respectively, 60 and 50. The lengths of the paths were 82.5 and 70.2 m, respectively. In the case of PDST [see Fig. 15], the required time in both cases was approximately 1 s. The total number of nodes that represented the entire set of trees were, respectively, 1355 and 1287. The total number of output path solutions were, respectively, 58 and 49. Finally, the lengths of the paths were 79.2 and 67.1 m, respectively. In these experiments, real data were used, and most of the solution paths were generated along the pavement, since there were low-dirt piles between the pavement and the unpaved road. However, some nodes were generated on the pavement and on the unpaved area that is flat enough to allow the vehicle to travel. The results indicate that some of the dirt piles were low enough to allow the vehicle to apply the condition of pattern 1 to break through.

In the case of the environment that includes well-organized roads, a very short processing time is required to find a solution path with fewer nodes in comparison to the simulation case described in the previous subsection. Consequently, the proposed motion planner also performed well in the organized outdoor environment.

X. CONCLUSION

This paper proposed a novel motion planner for off-road UGVs, which considers regulations depending on different situations, as well as the unique characteristics of each vehicle. The proposed motion planner can identify the solution within a feasible processing time, even in a relatively large-scale environment, because the state generation process of (x, y, ϕ) and (z, ψ, θ) which are the elements of each node is decoupled in a random sampling based manner. Note that an excessive computational time is required to search the 6-DoF elements within a large configuration space. The validity of the proposed motion planning scheme was investigated through a substantial number of experiments, both in a simulation environment with a rough terrain, and a real environment including organized pavements. The following conclusions were drawn.

1) The proposed motion planner considers the motion model of a vehicle and terrain information from OctoMap during random node sampling; thus, it is possible to generate a path reflecting the vehicle's moving ability.
2) Since a novel biased sampling scheme that generates control velocity data is applied that is based on an adaptive probability distribution in accordance to user-defined regulations, the motion planner used to generate an adaptive path that takes them into account works well.
3) A solution path that satisfied the vehicular characteristics while it avoided physical obstacles was produced because we designed the evaluation criteria based on vehicular characteristics and on the collision check in the configuration space.

Finally, the future work related to this study is as follows. In this study, the motion planner was implemented under the assumption that the map information of a target environment is pregiven. However, there is no guarantee that the pregiven map is an exact match to the actual environment. Moreover, vehicle localization error should also be considered when the vehicle follows the generated path. Therefore, we will develop a more robust motion planning scheme that could manage such uncertainties from map information and vehicle localization. In addition, we can take a new strategy to choose the best one of many path solutions into consideration. Given that our proposed planning scheme is able to identify a solution within feasible processing time, the planning process between the same initial configuration and target configuration can be repeated several times. Thereafter, a user can select one of the most appropriate paths based on a user-defined cost function that evaluates various physical quantities, such as length, smoothness, amount of work, and fuel consumption.

REFERENCES

[1] S. Choi, E. Kimm K. Lee, and S. Oh, "Real-time nonparametric reactive navigation of mobile robots in dynamic environments," *Robot. Auton. Syst.*, vol. 81, pp. 11–24, 2017.

[2] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auton. Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[3] C. P. Lam, C. T. Chou, K. H. Chiang, and L. C. Fu, "Human-centered robot navigation–towards a harmoniously human–robot coexisting environment," *IEEE Trans. Robot. Automat.*, vol. 27, no. 1, pp. 99–112, Feb. 2011.

[4] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *Int. J. Social Robot.*, vol. 8, no. 1, pp. 51–66, 2016.

[5] A. Y. Hata and D. F. Wolf, "Terrain mapping and classification using support vector machines," in *Proc. 6th IEEE Latin Amer. Robot. Symp.*, 2009, pp. 1–6.

[6] Y. Tanaka, Y. Ji, A. Yamashita, and H. Asama, "Fuzzy based traversability analysis for a mobile robot on rough terrain," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3965–3970.

[7] J. F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Terrain classification using three-dimensional ladar data for ground robot mobility," *J. Field Robot.*, vol. 23, no. 10, pp. 839–861, 2006.

[8] M. Bellone, A. Messina, and G. Reina, "A new approach for terrain analysis in mobile robot applications," in *Proc. IEEE Int. Conf. Mechatronics*, 2013, pp. 225–230.

[9] S. M. LaValle, "Rapidly-Exploring random trees, "A new tool for path planning," Comput. Sci. Dept., Iowa State Univ., Tech. Rep. TR 98-11, pp. 1–4, 1998.

[10] L. E. Kavraki, P. Svestka, J. -C. Latombe, and H. M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[11] A. M. Ladd and L. E. Kavraki, "Motion planning in the presence of drift, underactuation and discrete system changes," in *Proc. Robot., Sci. Syst.*, 2005, pp. 233–240.

[12] C. Crane *et al.*, "Team CIMAR's NaviGATOR: An unmanned ground vehicle for the 2005 DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 8, pp. 599–623, 2006.

[13] S. Thrun, "The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.

[14] M. Ono, T. J. Fuchs, A. Steffy, M. Mainone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *Proc. IEEE Aerosp. Conf.*, 2015, pp. 1–10.

[15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot.*, vol. 30, no. 7, pp. 846–894, 2011.

[16] J. Lee, C. Pippin, and T. Balch, "Cost based planning with RRT in outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 684–689.

[17] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-Time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Jul. 2009.

[18] C. Richter, J. Ware, and N. Roy, "High-Speed autonomous navigation of unknown environments using learned probabilities of collision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 6114–6121.

[19] D. Fassbender, A. Mueller, and H.-J. Wuensche, "Trajectory planning for car-like robots in unknown, unstructured Environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3630–3635.

[20] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 601–606.

[21] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2276–2282.

[22] S. Kim, J. Kang, and M. J. Chung, "Probabilistic voxel mapping using an adaptive confidence measure of stereo matching," *Intell. Service Robot*, vol. 7, no. 2, pp. 89–99, 2013.

[23] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, 2012.

[24] A. Horunge, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3d mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[25] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[26] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 3859–3866.

Yonghoon Ji (S'15–M'16) received the B.S. degrees in mechanical engineering and computer engineering from Kyung Hee University, Seoul, South Korea, in 2010, the M.S. degree in mechatronics from Korea University, Seoul, South Korea, in 2012, and the Ph.D. degree in precision engineering from The University of Tokyo, Tokyo, Japan, in 2016.

He was an Overseas Researcher under the Postdoctoral Fellowship of Japan Society for the Promotion of Science with the Department of Precision Engineering, The University of Tokyo, from 2016 to 2018. He is currently an Assistant Professor with the Department of Precision Mechanics, Chuo University, Tokyo, Japan. His research interests cover mobile robotics, unmanned vehicle technologies, and underwater robotics.

Dr. Ji is a member of ICROS, KROS, RSJ, JSME, and SICE.

Yusuke Tanaka (S'15) received the B.S. degree in precision engineering from The University of Tokyo, Tokyo, Japan, in 2015, and the M.S. degree in precision engineering from Graduate School of Engineering, The University of Tokyo, Tokyo, Japan, in 2017.

He is currently with the Toyota Motor Corporation, Toyota, Japan. His research interests cover mobile robotics and unmanned vehicle technologies.

Yusuke Tamura (S'05–M'08) received the B.E., M.E., and Ph.D. degrees in engineering from The University of Tokyo, Tokyo, Japan, in 2003, 2005, and 2008 respectively.

From 2006 to 2008, he was a Research Fellow with the Japan Society for the Promotion of Science. He was a Project Researcher with The University of Tokyo, from 2008 to 2012 and an Assistant Professor of Chuo University, Japan, from 2012 to 2015. He is currently a Project Associate Professor with the Department of Precision Engineering, The University of Tokyo. His research interests include human–robot interaction, remote control technology, and sports engineering.

Dr. Tamura is a member of the JSME and RSJ.

Mai Kimura received the B.E. and M.E. degrees in aeronautics and astronautics from Kyoto University, Kyoto, Japan, in 2005 and 2007, respectively.

She is currently an Associate Senior Researcher with the Robotics Group, Robot Technology Department, Products Development Center, IHI Corporation, Yokohama, Japan. Her research interests cover mobile robotics, unmanned vehicle technologies, and teleoperation technologies.
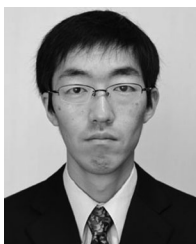
Atsushi Umemura received the B.E. degree in mechanical engineering from Waseda University, Tokyo, Japan, in 2004, and the M.E. degree in mechanical engineering from Waseda University, Tokyo, Japan, in 2006.

He is currently a Project Manager, Project Department, Technology Business Division, Silicon Studio Corporation, Tokyo, Japan. His research interests cover mobile robotics, unmanned vehicle technologies, and data visualization.

Yoshiharu Kaneshima received the B.E. degree in mechanical engineering from Waseda University, Tokyo, Japan, in 1999, and the M.E. degree in mechanical engineering from Waseda University, Tokyo, Japan, in 2001.

He is currently the Head of the Robotics Group, Robot technology Department, Products Development Center, IHI Corporation, Yokohama, Japan. His research interests include mobile robotics, unmanned vehicle technologies, and teleoperation technologies.

Hiroki Murakami received the B.S. and M.S. degrees in precision engineering from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively.

He is currently a Technical Advisor of Corporate Research & Development, IHI Corporation, Yokohama, Japan. His current research interests include remote-controlled vehicle for unmanned construction and industrial robot system.

Mr. Murakami is a Fellow of the Japan Society of Mechanical Engineers, the Institute of Systems, Control and Information Engineers, the Robotics Society of Japan, and a member of the Society of Instrument and Control Engineers.

Atsushi Yamashita (S'00–M'02) received the B.E., M.E., and Ph.D. degrees in precision engineering from The University of Tokyo, Tokyo, Japan, in 1996, 1998, and 2001, respectively.

From 1998 to 2001, he was a Junior Research Associate with RIKEN (Institute of Physical and Chemical Research). From 2001 to 2008, he was an Assistant Professor with Shizuoka University. From 2006 to 2007, he was a Visiting Associate with the California Institute of Technology. From 2008 to 2011, he was an Associate Professor with Shizuoka University. Since 2011, he has been an Associate Professor with the Department of Precision Engineering, The University of Tokyo. His research interests include robot vision, image processing, and intelligent sensing for robots.

Dr. Yamashita is a member of ACM, JSPE, RSJ, IEICE, JSME, IEEJ, IPSJ, ITE, and SICE.

Hajime Asama (M'87–F'18) received the B.S., M.S., and Dr.Eng degrees in engineering from The University of Tokyo, Tokyo, Japan, in 1982, 1984, and 1989, respectively.

He was with RIKEN, Japan, from 1986 to 2002 as a Research Scientist. He became a Professor of Research into Artifacts, Center for Engineering, The University of Tokyo in 2002, and has been a Professor with the School of Engineering, University of Tokyo, since 2009.

Prof. Asama was a recipient of the SICE System Integration Division System Integration Award for Academic Achievement in 2010, RSJ Distinguished Service Award in 2013, JSME Award (Technical Achievement) in 2018, etc. He was the Vice-President of RSJ in 2011–2012, an AdCom member of the IEEE Robotics and Automation Society in 2007–2009. He has been the president-elect of IFAC since 2017, the President of the International Society for Intelligent Autonomous Systems since 2014, and an Associate Editor for the *Control Engineering Practice*, *Journal of Robotics and Autonomous Systems*, *Journal of Field Robotics*, etc. He was a member of the Science Council of Japan, from 2014 to 2017, and has been a council member since 2017. He is a Fellow of JSME and RSJ.