

Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot

Dominik Belter, Przemysław Łabęcki, and Piotr Skrzypczyński

*Institute of Control and Information Engineering Poznań University of Technology ul. Piotrowo 3A 60-965 Poznań, Poland
e-mail: dominik.belter@put.poznan.pl, przemyslaw.labecki@cie.put.poznan.pl, piotr.skrzypczynski@put.poznan.pl*

Received 14 June 2014; accepted 17 April 2015

Achieving full autonomy in a mobile robot requires combining robust environment perception with onboard sensors, efficient environment mapping, and real-time motion planning. All these tasks become more challenging when we consider a natural, outdoor environment and a robot that has many degrees of freedom (DOF). In this paper, we address the issues of motion planning in a legged robot walking over a rough terrain, using only its onboard sensors to gather the necessary environment model. The proposed solution takes the limited perceptual capabilities of the robot into account. A multisensor system is considered for environment perception. The key idea of the motion planner is to use the dual representation concept of the map: (i) a higher-level planner applies the A* algorithm for coarse path planning on a low-resolution elevation grid, and (ii) a lower-level planner applies the guided-RRT (rapidly exploring random tree) algorithm to find a sequence of feasible motions on a more precise but smaller map. This paper contributes a new method that can identify the terrain traversability cost to the benefit of the A* algorithm. A probabilistic regression technique is applied for the traversability assessment with the typical RRT-based motion planner used to explore the space of traversability values. The efficiency of our motion planning approach is demonstrated in simulations that provide ground truth data unavailable in field tests. However, the simulation-verified approach is then thoroughly tested under real-world conditions in experiments with two six-legged walking robots having different perception systems.

© 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

Urban search and rescue (USAR) missions are demanding for autonomous robots (Murphy, 2014). The robot has to operate in a rough and risky environment that is usually unknown in advance. In such an environment, the “2D assumption,” which states that the robot moves on a flat surface and sensors take measurements in a plane parallel to that surface, does not hold anymore. This assumption is common for mobile robot navigation and motion planning and greatly simplifies these tasks. The motion planning problem becomes even more complicated if we consider not only a simple wheeled or tracked vehicle, but also an articulated robot that has many degrees of freedom, such as a legged robot. Recently, legged robots have been considered more often for the USAR missions, because of their high mobility and the ability to overcome both the rough terrain and structured man-made obstacles. In contrast to wheeled or tracked locomotion in which a robot adapts to the terrain shape by passive mechanical feedback, a walking robot has to actively adapt the footholds, gait pattern, and trajectory of each leg to the terrain. In a USAR mission, the walking robot should not only plan a motion sequence of its legs

Direct correspondence to: Dominik Belter, e-mail: dominik.belter@put.poznan.pl

to traverse the terrain and negotiate the obstacles but also create a model of the terrain and self-localize with regard to this model. All these functions have to be tightly integrated in order to ensure precise and reliable motion execution. Careful planning enables also the multilegged robot to better address the limitations of its own kinematics, such as the inability to freely choose the orientation of the feet.

Although some advanced six-legged robots rely on behavior-based control for rough terrain walking (Roennau et al., 2014), the pure behavior-based approach does not suit well a fully autonomous robot. Relying exclusively on the behavioral paradigm does not permit efficient, nearly optimal motion of the robot in a longer horizon, as the reflexes tend to result in repetitive, trial-and-error motion of the robot, and there is a possibility the robot will get stuck in a local minima or in an infinite loop of behaviors. However, the reactive behaviors may help to make the motion more robust (Roennau et al., 2009), compensating for the imperfection of the environment and robot models used in planning. Yamaguchi et al. (2013a) demonstrated that a robust motion policy for a walking machine can be obtained by reinforcement learning, without a precise environment model. Although such an approach does not permit an optimal motion plan, the success of learned policies suggests that a walking machine may also learn how to

compensate for unforeseen situations in the computed motion plans.

Therefore, our approach to rough terrain legged locomotion is based on the classic sense–plan–act paradigm, which is pretty effective providing that the whole control loop is executed precisely in real time. The robots used in our experiments are six-legged; therefore, statically stable gaits can be used. This simplifies to some extent the motion planning problem, comparing it with the dynamic gaits in quadrupeds or bipeds. However, in practical patrol or rescue missions, the additional stability provided by the six-legged configuration is very helpful, overcoming obstacles that are relatively large in comparison to the size of the robot. Our aim is to enable the walking robot to move to the given location autonomously, using only onboard sensors for perception, without any previous knowledge of the environment. We do not assume any particular terrain type or configuration of the obstacles.

1.1. Problem Statement

In this paper, we consider the problem of perception-based motion planning in rough terrain for a fully autonomous legged robot. It is assumed that the robot explores an unknown outdoor or indoor area, having no a priori map of this area. We do not use also any external system for positioning, such as external cameras or the GPS. For USAR missions, we want to have a robot that is capable of operating in GPS-denied environments, for example, inside buildings, in tunnels, or in the urban canyons made by tall buildings, where the GPS is not reliable enough. Thus, the environment perception is accomplished using only the onboard exteroceptive sensors, which have limited range and limited accuracy of measurements. As the robot operates in an unknown, GPS-denied environment, the operator has to specify the goal in relation to the current pose of the robot. It is intuitive for the human teleoperator to determine the goal, for example, 20 m in front of the robot, which might have meaning like “behind the wall.”

The terrain is modeled by means of elevation mapping, using a variant of the well-known grid-based elevation map. Because of the limited perception range, the environment mapping is incremental: as the robot proceeds toward the goal, which is set as a point in the global coordinate system, new areas of the terrain are discovered and mapped. Thus, the path planning problem can be solved only locally, and the path has to be replanned as the new information becomes available. However, the legged robot requires not only planning a geometric path to follow through the environment (like planning of a x , y , z path for a wheeled robot), but also planning the full-body motion sequence that enables the robot to overcome some of the encountered obstacles, for example, to select footholds and determine the pose of the robot’s platform in relation to the obstacles. It means that a walking robot should not only distinguish between

traversable and non-traversable obstacles but also find how to deal with the obstacle to exploit its walking capabilities. Moreover, because of the practical implementation of the motion planner on the real six-legged robots Messor and Messor II, we need to carefully take into account the properties of the perception systems of these robots. Both robots have multisensor perception systems. The Messor hexapod is equipped with the Hokuyo URG-04LX laser scanner used for precise but short-range terrain perception in front of the robot. Besides the laser scanner, the walking robot has a passive stereo camera for longer-range perception. The updated Messor II robot is currently equipped with only one exteroceptive sensor – the Asus Xtion Pro, which, however, provides both RGB images from a passive camera, and three dimensional (3D) range measurements from an active depth camera. Both configurations of sensors, although significantly different, share similar perceptual capabilities. The obstacles that are close to the current position of the robot can be mapped in detail, using either the tilted laser scanner or the Xtion depth camera. The data about environment features, which are farther from the robot, are much less accurate, because the range measurements provided by both the passive stereo and the active depth camera working in the farther part of its field of view are corrupted by noise and artifacts. Thus, the robot cannot perceive the terrain in detail in a longer horizon and is only capable of planning a rough path to a distant goal.

It is difficult to plan the whole motion sequence (footholds, leg trajectories, etc.) in advance. The full-body motion can be determined only for the laser-scanned local area. However, the general path to the goal (or to the limits of the known terrain map, if the goal is not visible from the current position of the robot) should be traversable with the given motion capabilities of the legged robot. Therefore, we need to determine the traversability of the terrain on the basis of the available elevation map and to assess the cost associated with the locomotion through the given area. We look for a method that predicts the possibility of finding a set of feasible motions of the hexapod to traverse the terrain, knowing only a rough elevation map of this terrain. Such a method should be integrated with the path planner to enable the full-body motion planner to robustly and efficiently follow the general path.

1.2. Approach and Contribution

Our approach to perception-based motion planning for a USAR walking robot resembles human behavior during a trip in rugged terrain: We usually plan a general path by looking at the area from a distance to avoid trees, large stones, and ditches and then we follow this path, focusing our visual attention on the terrain at our feet in order to avoid smaller obstacles and find appropriate footholds. These two planning levels are implemented in the robot by two different algorithms. The rapidly exploring random

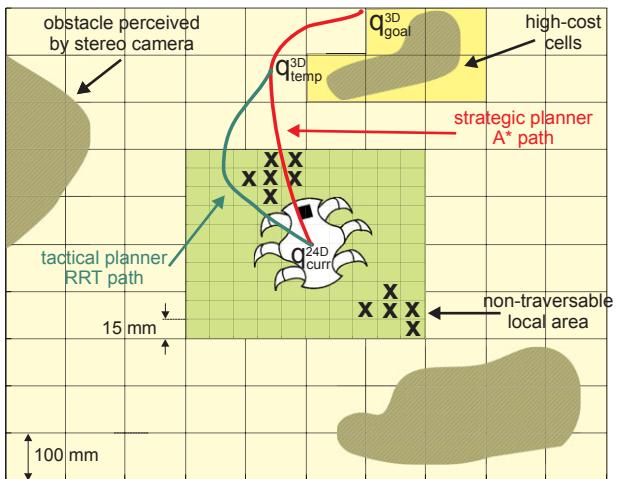


Figure 1. Concept of the motion planning approach and terrain representation – see text for explanation.

tree (RRT) algorithm (Kuffner Jr. & LaValle, 2000), which provides a single-step, probabilistically complete planning method for high-dimensional search spaces, is applied at the full-body motion planning level. At this level, the RRT-based planner handles the 6-DOF robot pose and the 18-dimensional space defined by the legs' joints. However, the probabilistic nature of RRT often leads to exploring spaces unrelated to any reasonable path. To mitigate this problem, the higher-level path planner is added, which uses the heuristic, nonrandomized A* algorithm. This algorithm works in a lower-dimensional search space, because the coarse elevation map representing farther areas allows only to roughly determine traversability between two cells. However, the A* algorithm is guaranteed to yield a path optimal up to the resolution and correctness of the available map. The path planned at the higher level is then used to define local subgoals for the lower-level RRT-based motion planner. Because the two planning levels run repeatedly as the robot moves and perceives new areas, the fine-grained motion plan yielded by the RRT follows the general path from the A* planner but can make detours if the coarse path turns out to be locally unfeasible for the walking robot.

The approach to motion planning is closely related to the environment representation employed by the robot. We use a dual-resolution terrain representation (Figure 1), which employs a larger, coarse elevation grid describing the more distant areas perceived by means of the stereo or RGB-D sensor in the far field, and a smaller, fine-grained local elevation map that surrounds the robot and is updated from the more precise and reliable range data from the laser scanner or the RGB-D sensor working in the near field. The robot is located in the center of these maps, and both maps are translated (but not rotated) as the machine is walking.

To implement a complete perception-based locomotion planning system for an autonomous legged robot, several subproblems have to be solved first. In our previous publications, we have presented in details the theoretical foundations and practical aspects of the foothold selection method (Belter & Skrzypczyński, 2011b), legs trajectories planner (Belter & Skrzypczyński, 2011a), and posture optimization algorithm (Belter & Skrzypczyński, 2012). The motion planning system presented in this paper builds up to those results, which are integrated as modules of the precise motion planner. The description of these modules given here (subsection 5.3) should allow the reader to understand the principles of their operation and how they are integrated within the planner, but readers interested in further details and implementation should consult the original papers. Moreover, in this work, we extend and unify our earlier concepts of multisensor terrain mapping (Belter et al., 2012) and dual-resolution terrain representation (Belter et al., 2011), enabling the motion planner to exploit real-time terrain perception and thus achieving autonomy.

This article presents a unified view of all the aspects of motion planning for the USAR hexapod robot that were previously tackled separately in our papers, describes in details the new traversability prediction method, previously only briefly introduced in the workshop paper Belter et al. (2013), and demonstrates the performance of our system on two real walking robots in extensive field experiments. The main new contribution of this paper is a method, which allows increasing the efficiency of motion planning using traversability prediction. We demonstrate that the efficiency of the motion planner depends on the traversability assessment, and then we show how to improve this efficiency by adaptation of the traversability cost to the specific terrain and robot. We use a realistic simulator to provide data samples for off-line computation of the traversability cost, instead of learning directly on the real robot, which is motivated by the availability of perfect ground truth in the simulator, and the need to protect the valuable walking robot from being broken during experiments.

The rest of this paper is organized as follows. In Section 2, we review the related work in motion planning for legged locomotion in rough terrain. An overview of the proposed navigation system's structure is given in Section 3. Section 4 gives an overview of the terrain mapping module of our system, focusing on the map representation issues and how it is related to motion planning. Section 5 provides a detailed description of the approach to motion planning, with a brief presentation of the supporting modules, responsible, for example, for foothold selection and posture optimization. Results demonstrating the ability of the system to navigate various terrains and giving insight in the performance of the proposed methods in a real, outdoor environment are provided in Section 7. Section 8 concludes the paper.

2. RELATED WORK

This section surveys the motion planning and terrain perception work done so far, which is directly related to our approach.

2.1. Environment Modeling

Environment modeling for motion planning in legged robots was pioneered by the work on Ambler (Krotkov & Hoffman, 1994), which used a 3D laser scanner to build an elevation grid of the terrain. Some of the grid-based mapping algorithms use ad hoc formulation for sensor fusion, for example, Ye and Borenstein (2004) apply simple heuristics when they fuse consecutive range measurements from a tilted-down 2D laser scanner on a wheeled vehicle into an elevation map. Their certainty measure is a hit/miss counter, and they do not address fusion of data from various sensors, focusing on the Sick LMS 200 scanner. In Belter & Skrzypczyński (2011b), we have presented a modified version of the Ye and Borenstein's CAS (certainty assisted spatial) mapping algorithm that is appropriate for a walking robot with the low-cost Hokuyo URG laser scanner yielding sparse 2D range measurements. We also demonstrated in our previous works that the modified CAS approach can work with other sensors perceiving the terrain: a simple structured light 2D sensor (Łabęcki et al., 2011) and a passive stereo vision camera (Belter et al., 2011).

Stereo vision was used for terrain modeling with legged robots in only a few projects that reached the stage of field tests, mostly because typical stereo systems impose a high cost of the 3D point cloud computation. This cost is often prohibitive for onboard processing in a legged robot. In Kolter et al. (2009) and Rusu et al. (2009), walking robots with stereo-based perception are shown to traverse autonomously rugged terrain, but in both cases, there is no explicit propagation of the spatial uncertainty from the stereo data to the elevation map. A stereo camera was also used in Stelzer et al. (2012) to obtain a grid-based terrain model for autonomous navigation of a wheeled or legged robot. Although this model stored, besides the elevation, some variables indicating the "age" of the height values (then used in the traversability estimation), no direct representation of the height uncertainty was employed. A passive stereo-vision-based terrain mapping system is also used in the recently published research on the Legged Squad Support System (LS3) quadruped vehicle (Bajracharya et al., 2013). As stated by the authors, a dense stereo system is used because it is the only sensor that provides the 3D data with the required resolution and within the size, weight, and power constraints of the LS3 robot. In this context, it should be noted that we introduced a method to efficiently combine the long-range dense stereo data and the near-field active 2D laser scanning data for terrain mapping in a much smaller legged robot (Belter et al., 2012). Moreover, we demonstrate here that an active RGB-D sensor is a practical

alternative to passive dense stereo for terrain perception under some environmental conditions (i.e., no direct sunlight) that are often encountered in USAR tasks (Murphy, 2014). The LS3 terrain mapping system addresses the problems of noise and artifacts in the dense stereo data in a manner similar to our approach, but we do not assume any particular type or structure of obstacles (like thin objects interpreted as vegetation) to be detected, because in the USAR tasks the geometry of the terrain may be arbitrary, complicated, and hard to interpret (e.g., a pile of debris with protruding wires). The grid-based terrain map used in Bajracharya et al. (2013) includes the surface elevation, local slope and roughness, and discrete positive/negative obstacles, but there is no mathematical uncertainty model underlying the mapping process. In contrast, we employ a rigorous probabilistic framework that enables us to avoid arbitrary threshold values in elevation updating and creates a common ground for data fusion from various sensors.

Although many recent systems do not estimate the uncertainty of elevation values in terrain maps for walking robots that motivate this by the need for real-time data processing, several approaches to estimate the elevation uncertainty have been proposed so far. The *locus* method of (Krotkov & Simmons, 1996) applies simple geometric approximation in order to obtain elevation errors from the sensor error model. This approximation, however, does not account directly for the uncertainty of range measurements. Recently, a method for efficient volumetric mapping was presented, which employs the notion of positive and negative occupancy mass to represent the environment (Dryanovski et al., 2010). Although this model is efficient with regard to the map size, it is rather suited for robots that require full 3D world model, not for walking robots, which can use much simpler elevation maps.

Terrain maps obtained from online measurements are employed also in motion planning and walking control for humanoid robots; however, these maps often are based on simplifying assumptions in order to achieve accurate mapping in real time on a highly dynamic and unstable platform. Cupec et al. (2005) assumed an environment, including only straight lines, while Chestnutt et al. (2009) simplifying the mapping problem assuming that the environment consists of horizontal plane patches. In the more recent work on the navigation of a humanoid robot in unknown environment (Nishiwaki et al., 2012), the terrain is represented by a more typical elevation grid. Although a rather precise 3D laser scanner is used for perception, the obtained terrain model is of limited size due to the possible mapping errors induced by imprecise self-localization, and there is no representation of the elevation uncertainty that may be used by the navigation algorithms.

Although the grid-based approaches to terrain mapping dominate in practical applications due to their simplicity and low computational requirements, it should be noted that entirely different concepts are also presented in

the literature. Plageman et al. (2009) apply a nonparametric continuous surface estimation method. This method is based on Gaussian process regression and applies nonstationary covariance functions for adaptation of the regression kernels to the structure of the terrain. It enables filling in large gaps in elevation data and provides estimates of uncertainty for the filled-in regions. However, these advantages come at the cost of high computational burden, which can be prohibitive for the embedded computer of a walking robot.

2.2. Traversability Assessment

There exist also some works that focus on adapting the environment representation for motion planning in rough terrain. The notion of variable-resolution grid map for motion planning is used in Kirby et al. (2009). The spatial resolution of the map decreases with the distance from the robot, which allows to use an optimal A* path planner in real time for an environment with moving obstacles. A similar grid map concept is applied in our work; however, it is motivated mostly by the properties of the perception systems of our robots. Our high-level path planner exploits some form of cost map computed over this variable-resolution grid. It is based on the spherical variance, which was introduced by Sanctis et al. (2009). They calculate a weight matrix based on the gradient of each point over the 3D surface, the spherical variance, and the robot limitations. This matrix is used to limit the speed propagation of the fast marching wave-propagating path planner in order to find the best path.

More generally, the terrain types might be classified using photometric or range images. Each terrain type has a property related to the robot's mobility capabilities. An extended survey on traversability analysis methods using various sensors can be found in Papadakis (2013). Shirkhodaie et al. (2005) proposed using three classifiers: rule based, neural network, and fuzzy logic. The best performance on Martian terrain classification was obtained by a fuzzy-logic classifier. Similar results are presented by (Howard & Seraji, 2001), who proposed extracting four features that describe terrain properties (roughness, slope, discontinuity, and hardness) from camera images using a fuzzy logic framework.

A robot can learn from previous experience how to assess terrain properties (Shneier et al., 2008). In this case, the traversability assessment algorithm uses both range and visual data. While traversing rough terrain, the robot refines the traversability cost map. In contrast, we avoid online learning on the physical robot employing a simulator. Yamaguchi et al. (2013b) also demonstrate that the behavior-based policy for generating motion may be adapted to various terrains by learning from previous experience. The trial-and-error exploration of a terrain with unknown features just for learning the cost map, as proposed by Shneier et al. (2008), might be dangerous for a real legged

machine. A wheeled robot uses mechanical bumpers to detect nontraversable regions, while a legged robot easily can fall down or tip over and break while negotiating a nontraversable obstacle of arbitrary shape. Moreover, in a robot using off-the-shelf servos, like Messor and Messor II, a prolonged trial-and-error repetitive behavior may lead to overheating and damage to the servomotors. Such treads motivate the use of simulators in the systems' evolving legged robot gaits (Belter & Skrzypczyński, 2010) and also should be considered in the task of traversability assessment.

In the traversability assessment, taking into account the local geometry of the terrain, the traversability cost for a wheeled vehicle can be simply computed as a function of the slope of the terrain (Ye & Borenstein, 2004), but an extended vector of the geometric features (slope, roughness, step height) and some hand-tuned heuristics have to be used for a walking robot, like in the DLR Crawler robot (Stelzer et al., 2012). The purely geometric approach ignores such important properties of the terrain as friction, softness, and the ability to support the robot's mass. However, these properties cannot be directly observed by the currently available exteroceptive sensors and can be only obtained by haptic probing of the ground (Hoepflinger et al., 2014) or inferred from the semantic labels (e.g., snowy/muddy field or slippery floor) attached to particular areas in the process of terrain classification (Walas & Nowicki, 2014).

In this paper, we show that the heuristic given by a human expert is not always the best solution for the mobile robot. We show that the robot can collect the reference samples about the traversability of the terrain and then use this data to obtain more general knowledge about assessment of the terrain traversability. We use a mixture of Gaussians to obtain general relation between the terrain properties and the cost of motion. Our regression-based approach to terrain traversability assessment is directly motivated by the successful foothold selection method we have presented in Belter & Skrzypczyński (2011b). This method also used a realistic simulator to learn the dependency between some geometric characteristics of the terrain and the desired behavior of the walking machine; however, the terrain assessment proposed in Belter & Skrzypczyński (2011b) was specific to the problem of selecting acceptable footholds in the vicinity of the robot.

Among the related works in traversability assessment for mobile robots, the one by Papadakis and Pirri (2012) bears the highest similarity to our approach. It describes a simulation-based model that quantifies 3D traversability by accounting for intrinsic robot characteristics and articulating capabilities together with terrain characteristics. However, in contrast to our approach for walking robots, in Papadakis and Pirri (2012), an articulated tracked vehicle is considered, which leads to some simplifying assumptions that we cannot make. Moreover, the relatively small number of the robot's DOF implicitly assumed by Papadakis and Pirri (2012) makes it possible to find the

optimal configurations by exhaustive search in simulation. The search space of the possible solutions (configurations) for the six-legged machine we consider is too large for such an approach; thus, we employ the randomized RRT algorithm to explore the search space and find feasible configurations.

2.3. Motion planning

There are many approaches to motion generation in legged robots. A robot can walk over rough terrain without any representation of the environment and explicit planning of motion sequences. The RHex robot is designed to adapt to the terrain by using the specific mechanical design of its legs (Rusu et al., 2009). Mechanical adaptation to terrain irregularities is simple and reliable, but such a robot can damage the surfaces or objects it is moving on, which is undesirable in many applications, including USAR missions.

The high number of combinations of the possible robot's postures and states of the legs in a multilegged robot makes the motion planning a difficult and computationally expensive task (Arain et al., 2013). Therefore, some researchers try to simplify the problem. Such a radical simplification can be achieved by adapting existing 2D path planning algorithms to operate in rough terrain, assuming that the general 2D path can be robustly followed using a reactive controller. For example, Bai and Low (2001) mark obstacles on a legged robot's path as inaccessible or traversable and use a potential field algorithm for path planning. This simplification paradigm was used also on the BigDog robot (Wooden et al., 2010). A 2D cost map was created over the perceived terrain. The cost was high for detected objects and their neighborhood and lower for traversable areas. Then an optimal path was searched by using the classic A* algorithm. Then, traversing the moderately rugged terrain along this path is possible owing to the BigDog's robust reactive motion controller. The successor of BigDog, the LS3 quadruped, employs a stereo-vision-based online terrain mapping method, which creates a hybrid 2D/3D representation of the environment. This mapping system has been tested extensively in the field experiments (Bajracharya et al., 2013), but at the reported stage of the research, the robot wasn't autonomous and did not plan its motion nor even the path. It was teleoperated, while the terrain map was used merely to adjust few parameters of the cyclic gait executed by the robust controller. The terrain map enables the LS3 to predict the need for gait adjustment, increasing the robustness with respect to the purely reactive trial-and-error approach, but the robot still requires steering from a human operator. While this is acceptable (or even desirable) for a military vehicle, in USAR missions full autonomy for extended time periods may be required to cope with the limited wireless communication issues. In an attempt to reuse some 2D planning techniques, Kolter et al. (2008) divide the motion planning problem into two

layers. They at first plan a path of the LittleDog robot trunk without considering the footholds or trajectories of the feet. Then, the footholds are planned to follow the initial path while considering appropriate constraints. This approach may generate quite conservative motion plans, which do not take full advantage of the locomotion capabilities of a legged robot. In a two-stage planner, some feasible solutions could be eliminated in the early stage of planning.

Some planning algorithms, like the A*, that are popular in planning paths for wheeled robots in 2D maps have been directly adopted for computing paths of articulated (but not legged) robots in rough terrain. Bonnafous et al. (2001) integrated A* with a method that generates elementary motions on rough terrains, explicitly taking into account the geometric constraints of a wheeled articulated robot. In a recent paper (Norouzi et al., 2012), a search algorithm based on A* generates a path for a reconfigurable tracked robot that guarantees stability while at the same time maximizes the height of the sensor payload. The A* search algorithm is used in the classic approach to footstep planning for humanoid robots (Chestnutt, 2013). This approach involves a set of possible foot positions (stances) that is then explored by the A*-based planner. This set can be optimized locally in order to adapt to the terrain. Nishiwaki et al. (2012) demonstrate the use of A* search to generate a sequence of footstep locations for the HRP-2 humanoid robot. The footstep planner is integrated with online terrain mapping from a laser scanner data and a robust walking controller with sensor feedback from the ground reaction forces. A related work by Vernaza et al. (2009) presents the R* planning algorithm, which combines aspects of deterministic and randomized search. Hornung et al. (2012) demonstrate the use of R* and ARA* (another variant of A*) for footstep planning of a simulated humanoid robot. These algorithms, based on inflated heuristics, are usually faster than the basic A* version but are prone to fail in the local minima of the search space and are much more dependent on the underlying heuristics, which is particularly hard to obtain in the problem of rough terrain walking. A too simple heuristics that does not take into account all the constraints and motion possibilities of a legged robot may severely misjudge the path cost, leading to search failure. In such an approach as in Hornung et al. (2012), the efficient use of A* variants is possible with a small set of allowed footstep actions of the humanoid. In a six-legged robot walking over a rough terrain, the search space is much larger, due to the high number of DOF (in the humanoid case only DOF in the legs are considered), the possible use of a free gait that does not have fixed steps and a terrain model that is usually more complicated than the discrete and rather shallow obstacles considered in most of the experimental work with humanoids. A navigation algorithm, which bears some similarity to our approach to the coarse path planning, is presented in Stelzer et al. (2012). This algorithm uses the D*-Lite planner, which enables fast replanning. Similar to our system the approach of Stelzer

et al. (2012) employs stereo vision for online terrain modeling and estimates the traversability, but it does not provide precise planning of footholds and leg trajectories, as it is conceived as being suitable for either wheeled or legged robots.

Employing a sampling-based planning strategy is a way to tackle the problem of high dimensionality in motion planning for legged robots. Two frameworks are commonly applied for sampling-based motion planning: the probabilistic roadmap method (PRM) (Kavraki et al., 1996) and the rapidly exploring random tree (LaValle & Kuffner, 2001). Hauser et al. (2006) successfully used a PRM-based technique to plan motion of a six-legged lunar robot. The PRM framework has been also applied to plan nongaited, quasi-static walking motion of a humanoid robot over rough terrain (Hauser et al., 2008), resulting in, however, very long planning times (several minutes per single step of the robot).

Another sampling-based, probabilistically complete planning algorithm is the RRT. This algorithm creates a tree in the configuration space, with the nodes representing feasible, collision-free configurations and edges representing feasible actions that allow the robot to reach these nodes (LaValle, 1998). The RRT gained popularity as a framework for motion planning in various subfields of robotics, mostly due to its efficiency with high-dimensional problems and the ability to respect the kinematic and dynamic constraints existing in the problem (Sanchez et al., 2008). The greedy RRT variant can't find the solution when the configuration of obstacles is difficult. The RRT-Connect algorithm (Kuffner Jr. & LaValle, 2000) improves the efficiency by growing two trees simultaneously: one from the start and another one from the goal. Each tree expands toward the recently added node of the other tree until they connect.

Some RRT improvements consider a bias of the direction to the goal (Lindemann & LaValle, 2004). The RRT-path algorithm proposed in Vonasek et al. (2009) employs an auxiliary path (visibility diagram or Voronoi diagram) that guides the tree through the environment. The tree is attracted to grow along the auxiliary path from the start to the goal position of the robot also through narrow passages in the configuration space. This approach was to some extent an inspiration for our two-level motion planner; as in our case, the A* path guides the RRT algorithm through the map of the rough terrain.

The RRT-based planners have been already applied for planning paths in rough terrains, for example, on Lunar and Mars rovers (Kubota et al., 2001). In RRT implementations, the expansion of the tree is usually based on the Euclidean distance to the goal. However, a map of rough terrain is better represented as a cost space (cost map), and some RRT-based path planners directly consider the cost of transition over the terrain (Jaillet et al., 2008). Owing to this improvement, the computed paths efficiently follow low-cost valleys of the cost map. Lee et al. introduced the metric adaptive RRT, which considers the underlying cost of a path

when calculating the distance function for tree expansion, demonstrating it on a real wheeled robot (Lee et al., 2008). The RRT paradigm has been used to plan motion of various legged robots. A planner for statically stable motion of a humanoid robot in the presence of obstacles has been shown in Kuffner et al. (2002). In the more recent work of Perrin et al. (2012), an RRT planner is used to find collision-free sequences of footsteps for the HRP-2 humanoid avoiding 3D obstacles on a flat ground. The RRT is applied instead of the classical A* search in order to deal with a transition model, which is much larger than a relatively small set of possible steps used typically in the A*-based planning approaches. A method for planning statically stable and collision-free gaits of the RoboSimian quadruped robot over uneven terrain proposed in the very recent work by Satzinger et al. (2014) is based on the RRT-Connect algorithm but incorporates some heuristics to cope with the redundant kinematics of the RoboSimian. In contrast, the work of Shkolnik (2010) considers the problem of planning dynamic, underactuated motion of the LittleDog quadruped in very uneven terrain. A modified RRT-based planner is proposed, which uses motion primitives to reduce the size of the space of possible actions, and samples configurations from a subset of the full state space related to the task being considered. This planner is able to handle challenging dynamic constraints associated with the underactuated motion and was demonstrated to achieve bounding motion over steps and logs with the real LittleDog robot. Although these results are impressive with regard to the dynamics of motion, the LittleDog robot was modeled as a planar five-link system in order to reduce the search space for planning. Therefore, the motion was constrained to 2D, moving left and right legs together. Moreover, all the experiments were implemented using a known terrain map and the Vicon motion capture system for precise positioning.

The general approach to motion planning presented in this paper is similar to the method used on the LittleDog robot by Kalakrishnan et al. (2010). They use a hierarchical controller to plan and execute motion sequences of the robot. The robot is taught off-line how to select footholds. Expert demonstration employing a template-based learning algorithm is used. The robot trunk path is planned in regions with good footholds. Then, the motion controller executes the desired path, avoiding collisions and optimizing the trajectories of the legs. Moreover, simple reactions from force sensors located in robot's feet and the inertial measurement unit (IMU) are implemented. This allows the compensation of inaccuracies of the environment modeling and some robot localization errors. Although this approach has been shown to be efficient in real experiments with the dynamically stable quadruped robot, it should be noted that the hierarchical controller gives no possibility to re-plan the motion sequence if it turns out that the coarse path is infeasible, for example, due to imprecise terrain map. Therefore, it is not directly applicable to our problem of

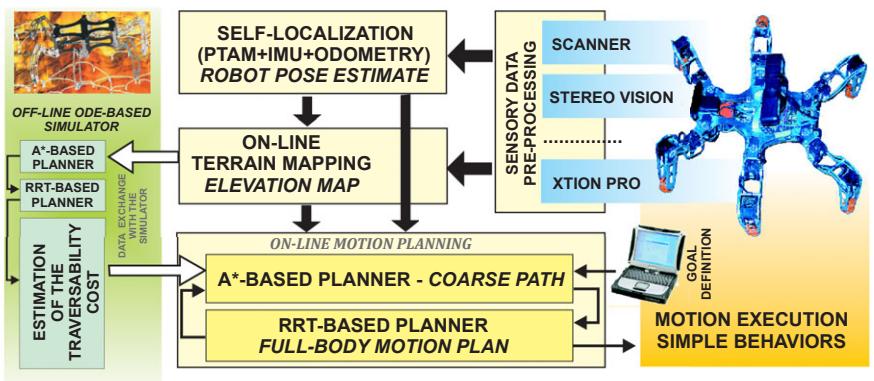


Figure 2. Structure of the navigation system for autonomous rough terrain traversal.

perception-based motion planning, where the adaptability of the planner driven by real-time perception is essential.

According to our best knowledge, we were the first to show an RRT-based motion planner for a multilegged autonomous robot traversing rough terrain, not only in simulation but also on the real Messor hexapod (Belter & Skrzypczyński, 2011a; Belter et al., 2011). Our approach to motion planning falls into the category of sampling-based methods, using the RRT concept, but avoids decomposition of the motion planning problem into several stages. Instead, the issues specific to a walking robot, for example, foothold selection and stability maintenance are solved by specialized subroutines within the main RRT-based planner. Thus, this approach is called “integrated.” The full-body motion planner is guided by the A*-based path planner, which significantly improves the computation efficiency, avoiding unnecessary growth of the motion plan tree. Moreover, the cooperation of the two planning algorithms helps to cope with the limited perception capabilities of our autonomous robot by allowing the robot to have a feasible path over the roughly mapped extended area, and then allowing the improvement of this plan locally (to the full-body motion plan level) whenever the more precise local map becomes available.

3. OVERVIEW OF THE NAVIGATION AND MOTION PLANNING SYSTEM

The motion planning algorithms presented in this paper are an integral part of a software system for fully autonomous navigation of a multilegged robot over rough terrain. This system consists of several modules that exchange data and control commands, as shown in Figure 2. The main functional modules are the motion planner, the terrain mapping module, and the self-localization module. In this paper, we focus on the two former subsystems, while details of the latter one (i.e., self-localization) are beyond the scope of this

work. Currently, we have adopted the software developed in our earlier research (Belter & Skrzypczyński, 2013) as the self-localization module, although we consider developing a precise self-localization method that is suited for a walking robot and is reliable under various environmental conditions, still an open issue.

To make the legged robot autonomous, the environment sensing is accomplished online using depth sensors (either a passive stereo camera, Asus Xtion Pro RGB-D camera, or a laser scanner). The mapping subsystem uses the acquired depth data to provide a terrain model (elevation map) to the motion planning module. Then, the obtained map is used to plan the motion of the robot. First, a coarse path to the goal is found using the A*-based planner. Then, full-body motion plan for the robot is determined by the RRT-based motion planner, which is guided by the coarse path. The motion plan is executed by the robot controller.

The controller basically takes the simple position-based approach when executing the motion commands, but using only the position-based control is risky on a rough terrain. Therefore, the robot uses the contact sensors in its feet to detect collisions with the ground and to stop the movement when support of the terrain is sufficient. Also when the real inclination of the terrain differs from the measurements, the robot changes the distance of the platform above the ground to increase the range of its legs. When the terrain is below the expected level, the robot moves body down. If the real terrain level is higher than the level given by the map, the robot moves the body up. These reactive behaviors significantly increase robustness of the motion execution. Currently, the reactive component of our system consists of few behaviors that are hand-crafted, taking into account our experience from experiments with real robots. Although it is possible to learn or adapt walking behaviors using previous experience of the robot (Yamaguchi et al., 2013b), we do not tackle this problem here and consider this a future research topic.

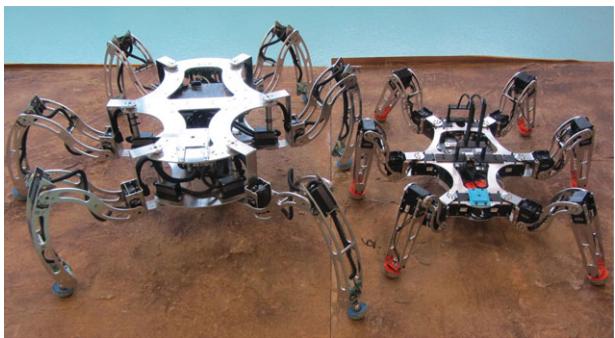


Figure 3. Legged robots: Messor (left, without exteroceptive sensors) and Messor II (right).

An important part of the approach presented in this paper is the new method for estimation of the terrain traversability cost. The parameters of the traversability cost estimator are found off-line using a realistic simulator of the walking robot. This simulator is fed with the terrain maps produced by the navigation system, thus the cost can be adapted to newly observed terrains. In the simulation environment, a somewhat simplified version of the motion planner is implemented, as the RRT-based planner is used as a reference when exploring a new terrain type (see Section 6). The online motion planner uses the optimized cost estimator to determine the traversability cost for the A* algorithm.

The navigation and motion planning system has been tested extensively on two mid-sized six-legged robots: Messor and Messor II. These robots are presented in Figure 3. The robots are research platforms that can carry various types of sensors. Both robots have the same insect-like kinematic structure. The Messor II robot is about 30% smaller than the older Messor robot. Moreover, Messor II is equipped with stronger servomotors. Thus, the total weight that might be carried by Messor II is about 1.5 kg, while the total payload of the Messor robot is less than 1 kg. The trunk of the Messor II robot is 30 cm long and 20.5 cm wide. The coxa, femur, and tibia segments in each leg of this robot have the lengths: 4.9 cm, 12 cm, and 17.4 cm, respectively. The Messor robot is 38 cm long and 29 cm wide. The segments in each leg have the following lengths: 5.5 cm, 16 cm, 20.3 cm. The total masses of the Messor and Messor II robots are 4 and 2.5 kg, respectively.

The low-level control system of the Messor robot is based on several ARM7 microcontrollers and a PandaBoard computer used for higher-level kinematics and gait computations. The Messor II robot is equipped only with a more powerful PandaBoard ES computer. This approach eases development of the control software, because modifications of the controller require software changes only. In-depth descriptions of Messor and Messor II can be found in Walas et al. (2008) and Belter and Walas (2014), respectively.

4. TERRAIN PERCEPTION AND MODELING

The basic terrain mapping algorithm used here on the Messor and Messor II walking robots was detailed in our previous work (Belter & Skrzypczyński, 2011b). This algorithm is an adaptation of the Ye and Borenstein's method (Okubo et al., 2009) but is more suitable to a walking robot with the compact Hokuyo URG-04LX laser scanner. However, to efficiently plan the robot's body path while perceiving the terrain, a sensory system with much longer range than the tilted URG-04LX is needed. On the Messor robot (Figure 4(a) and (b)) we employ the Videre Design STOC (STereo On a Chip) passive stereo vision camera, which yields a dense depth map in real time, owing to the hardware-embedded image processing. Unfortunately, the STOC camera range data are sparse and corrupted by various artifacts. Therefore, on the new Messor II robot (Figure 4(c) and (d)) we use the active 3D range camera – Asus Xtion Pro, which is compact and lightweight. Currently, Messor II does not have a 2D laser scanner, as the Asus Xtion range measurements are precise enough to enable fine-grained grid mapping for foothold selection.

Taking into account the limitations of the sensors we use on our robots, we conceived a dual-grid elevation map (Belter et al., 2011), which employs a small, precise grid that surrounds the robot and is updated from all the range data available to the robot and a bigger grid describing the more distant areas, which on the Messor robot are perceived only by means of the STOC camera. However, in Belter et al. (2011), the data from stereo vision and 2D laser were fused by ad hoc rules, which didn't take advantage of the uncertainty models of the particular sensors. In Belter et al. (2012), we described a more rigorous treatment of the spatial uncertainty for elevation map estimation. This framework for grid-based elevation mapping is used in this research as the terrain perception and modeling part of the motion planning system for autonomous walking over rough terrain.

4.1. Sensors and Spatial Uncertainty

To acquire the terrain profile, the older Messor robot mainly uses the tilted down Hokuyo URG-04LX. The surface in front of the robot is scanned while walking. According to the manufacturer the URG-04LX scanner measures ranges up to 1 m with the accuracy of 10 mm and ranges between 1 and 4 m with the accuracy of 1% of the measured distance, but this specification is valid only for a white sheet of paper. The URG-04LX suffers from a strong dependency between the accuracy of the range measurements and the optical properties of the target surface because of the used ranging principle.

Rigorous modeling of measurement uncertainty, for example, obtaining the variance of the measured ranges is problematic for the compact scanner under study. As shown by others (Okubo et al., 2009), the standard

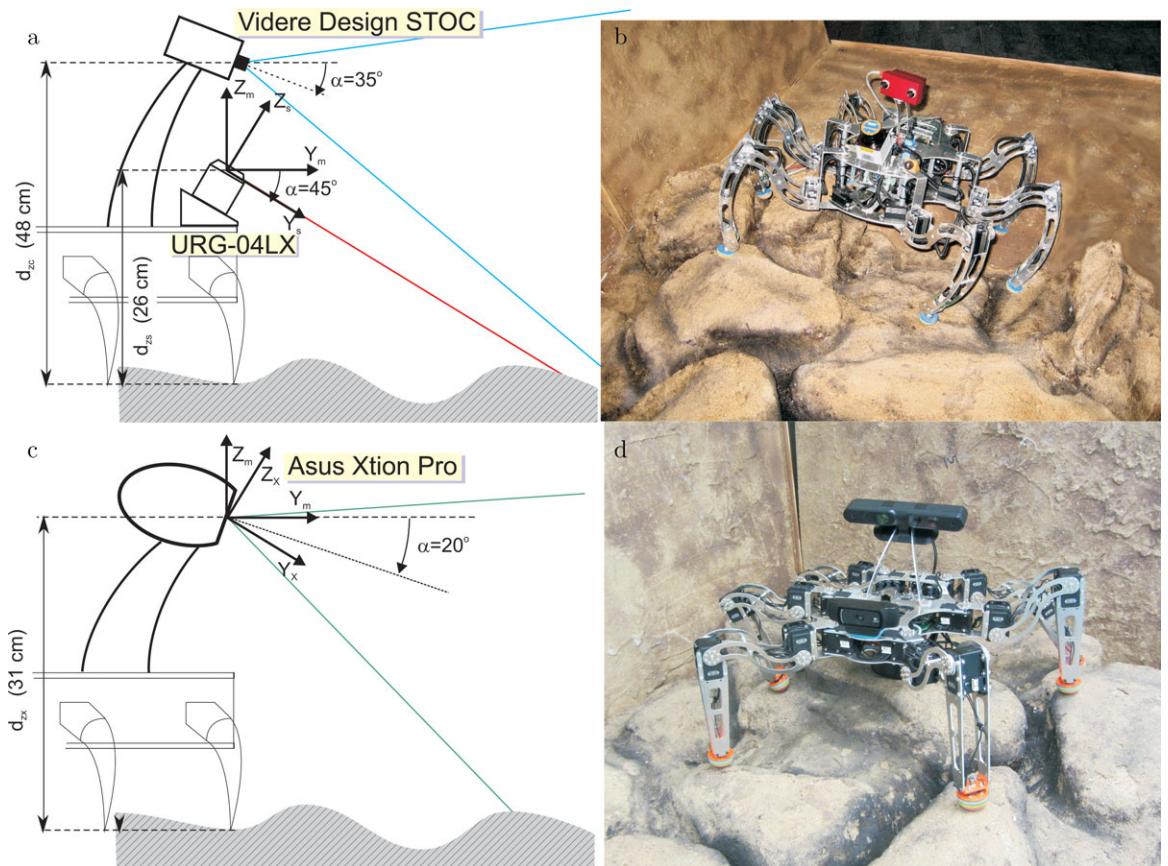


Figure 4. Geometric configuration of the terrain perception sensors on the Messor (a) and Messor II (c) robots, and the robots Messor (b) and Messor II (d) traversing rough terrain.

deviations observed for URG-04LX measurements are small even for large absolute errors in the measured distance. Eventually, we take a rough but conservative approximation of the range errors. We used experimental data for measured distances from 50 to 1000 mm with five different surfaces and the incidence angle of 30° . We picked the largest standard deviation for every measured distance and then established a mathematical model that approximates the change of the range standard deviation σ_r as a function of the measured range (Belter et al., 2012).

For updating the terrain map, each range r_s measured at the bearing angle β of the laser scanner is transformed to the elevation map coordinates \mathbf{p}_m^s by using the robot pose estimate:

$$\mathbf{p}_m^s = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \mathbf{T}_s^m \begin{bmatrix} r_s \sin \beta \\ r_s \cos \beta \\ 0 \end{bmatrix}, \quad (1)$$

where the homogeneous matrix \mathbf{T}_s^m defines the transformation from the laser scanner frame (denoted by the subscript s) to the coordinates of the grid (denoted by the

superscript m). This transformation depends on the pose of the walking robot, as the laser scanner is rigidly attached to the robot (see Belter & Skrzypczyński (2011b) equations (2) and (3)). The 6-DOF robot pose $\mathbf{x}_R = [x_R, y_R, z_R, \theta_R, \phi_R, \psi_R]^T$ contains the position in three dimensions and the yaw (orientation), pitch, and roll angles. The 3D points that update the map are then treated in the same way no matter what was the source of measurements, thus only the m (for map) subscript is used in the coordinates of a point. The spatial uncertainty of a point \mathbf{p}_m^s that updates the elevation map is computed as:

$$\mathbf{C}_{p_m^s} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} = \mathbf{J}_{RS} \mathbf{C}_{R(k)} \mathbf{J}_{RS}^T + \mathbf{J}_S \mathbf{C}_S \mathbf{J}_S^T, \quad (2)$$

where \mathbf{J}_{RS} and \mathbf{J}_S are the Jacobians of (1) with respect to (w.r.t.) the robot pose \mathbf{x}_R and the measurement r_s , respectively, $\mathbf{C}_{R(k)}$ is the covariance matrix of the 6-DOF robot pose at the k -th time instant when the range measurement was obtained, and \mathbf{C}_S is the range measurement covariance matrix, which

reduces to the range variance σ_r^2 according to our model (Belter et al., 2012).

To obtain information on more distant objects, the Messor robot is equipped with a stereo camera, which can work indoors and outdoors in various light conditions and provides data for a range of several meters. The Videre Design STOC camera implements in its hardware the small vision system (SVS) algorithm (Konolige, 1997), which uses correlation-based matching on rectified images, producing a dense depth map. The camera software outputs disparity images or, alternatively, point clouds. The stereo camera often produces incorrect matches. Errors occur as a result of insufficient image texture and around depth boundaries or when objects are too close to the camera. The artifacts on the borders of objects with disparity values are much different from the background manifest, as halos stretched from the foreground to the background object. Other artifacts, such as out-of-horopter measurements and measurements in textureless areas manifest themselves as groups of points near the camera. In Łabęcki (2011), we show how to remove the artifacts belonging to both groups by analyzing the depth image.

The sources of measurement errors in stereo vision are the finite image resolution and false matches. Because the matches are performed on rectified images and the rectification algorithm uses a calibrated camera model, the uncertainty of calibration also contributes to the overall measurement uncertainty. Matching errors are, due to their random nature, difficult to determine and thus are not considered here. The calibration uncertainty and the uncertainty resulting from the finite image resolution are propagated to the uncertainty of the pixel coordinates on the rectified image from the camera. This uncertainty is then propagated to the depth measurement uncertainty, which in turn is used to compute the uncertainty of the locations of the 3D measurements \mathbf{p}_c represented as a point cloud. A thorough description of the uncertainty assessment and propagation in the STOC sensor is given in Łabęcki and Skrzypczyński (2013). The coordinates of a point that updates the elevation map are calculated as:

$$\mathbf{p}_m^c = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \mathbf{T}_s^m (\mathbf{T}_c^s)^{-1} \mathbf{p}_c, \quad (3)$$

where \mathbf{T}_c^s is the homogeneous matrix of the transformation from the laser scanner to the camera coordinate system (denoted by subscript c), and \mathbf{p}_c is the vector that describes the measured 3D point in the camera coordinates. As the scanner-camera mounting on the Messor robot is rigid, we assume that the off-line calibration (Łabęcki & Belter, 2013)

is precise enough to neglect its uncertainty. Therefore, the uncertainty of a point in the map coordinates is given by:

$$\mathbf{C}_{p_m^c} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} = \mathbf{J}_{rc} \mathbf{C}_{R(k)} \mathbf{J}_{rc}^T + \mathbf{J}_c \mathbf{C}_{p_c} \mathbf{J}_c^T, \quad (4)$$

where \mathbf{C}_{p_c} is the covariance matrix of the 3D coordinates of point measured by the STOC sensor, \mathbf{J}_{rc} and \mathbf{J}_c are Jacobians of (3) w.r.t. \mathbf{x}_R and point coordinates \mathbf{p}_c , respectively, $\mathbf{C}_{R(k)}$ is the covariance of the robot pose at the time instance k when the stereo measurement was taken.

The Asus Xtion Pro used on the Messor II robot is based on the PrimeSense structured light camera, the same that is used in the more popular (but bigger and heavier) Kinect sensor. Therefore, we adopted for our purposes the mathematical uncertainty model for the spatial measurements developed for the Kinect (Park et al., 2012; Khoskelham & Elberink, 2012). This model describes the propagation of uncertainty between the disparity image and the Cartesian coordinates of the perceived points in the 3D space. This results in the covariance matrix of the measurement error, which represents the uncertainty for spatial position of features observed by the Kinect.

On the Messor II robot, the Asus Xtion Pro is attached to the robot in a similar way as the camera, tilted slightly downward to increase the terrain area in front of the robot that is in the field of view of the sensor. The Xtion Pro serves as the only range sensor, updating both the local precise elevation map, and the bigger coarse map. It is not necessary to transform the measurements between the coordinate frames of the sensors, like in the multisensor system on the Messor robot. Thus, the calculations of the coordinates of a point that updates the elevation map are simpler:

$$\mathbf{p}_m^x = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \mathbf{T}_x^m \mathbf{p}_x, \quad (5)$$

where the homogeneous matrix \mathbf{T}_x^m defines the transformation from the Xtion Pro frame (denoted by subscript x) to the coordinates of the elevation map, which depends on the pose of the robot, like in the case of the laser scanner on the older Messor. The uncertainty of the location of a measured point updating the elevation map is computed as:

$$\mathbf{C}_{p_m^x} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} = \mathbf{J}_{rx} \mathbf{C}_{R(k)} \mathbf{J}_{rx}^T + \mathbf{J}_x \mathbf{C}_{p_x} \mathbf{J}_x^T, \quad (6)$$

where the covariance matrix \mathbf{C}_{p_x} is obtained from the Kinect model (Park et al., 2012), while \mathbf{J}_{rx} and \mathbf{J}_x are the Jacobians of (5) w.r.t. \mathbf{x}_R and coordinates of a point in the sensor frame \mathbf{p}_x , respectively. As previously, $\mathbf{C}_{R(k)}$ is the covariance of the robot pose.

In contrast to reactive locomotion on rough terrain, the sense–plan–act paradigm underlying our approach to rough terrain traversal requires a precise self-localization

system. Without information about the pose of the robot, the obtained elevation map is distorted. Motion planning and path execution using such a map are almost impossible. For that reason, we use camera images and the parallel tracking and mapping (PTAM) algorithm (Klein & Murray, 2007) to localize the robot with respect to the global coordinates. As we have investigated earlier, the self-localization accuracy required for successful elevation map updates is at the level of the map cell size (Belter & Skrzypczyński, 2013). Such an accuracy (i.e., errors below 2 cm) is achievable with the bundle-adjustment-based PTAM, which operates in real time. The high accuracy is needed only within the local map used for full-body motion planning. The bigger map used for coarse path planning tolerates robot pose errors up to 10 cm. The images used by PTAM can come either from a separate monocular camera installed on the robot only for that purpose or from the STOC stereo sensor – in such a case the grayscale images from the left camera of the STOC sensor are streamed to the robot together with the disparity images used for 3D terrain perception. It should be noted that the 3D measurements from the STOC camera cannot be used for reliable self-localization due to their low quality at higher distances and the numerous artifacts that can cause false matches in the visual odometry or simultaneous localization and mapping algorithm. The point clouds obtained from the Xtion Pro have generally much better quality, providing that the lighting conditions are appropriate. We developed a visual odometry algorithm that uses RGB-D data efficiently (Nowicki & Skrzypczyński, 2013); however, this software is still not integrated on our Messor II robot. Therefore, currently we use PTAM on both walking robots involved in the research presented here. If the environment does not contain enough pointlike visual features for PTAM or the features cannot be matched reliably (at least 100 successfully matched points is needed), the robot uses legged odometry and the 6-DOF Attitude and Heading Reference System (AHRS) – XSense MTi. The decision when to switch to the legged odometry is taken on the basis of some PTAM internal variables that indicate whether the tracking of the camera (i.e., the robot) is reliable. The legged odometry provides means for short-time self-localization and allows correct elevation map updating whenever data from the visual odometry are temporarily not available. Moreover, our walking robots use contact sensors in each foot to reduce the influence of discrepancies between the terrain map and the real environment that may result from localization errors.

4.2. Terrain Mapping Method

In perception-based walking, the crucial problem is foothold selection, because any wrongly positioned step can result in falling down (Belter & Skrzypczyński, 2011b). The foothold selection algorithm requires a high-resolution elevation map that captures geometry of the terrain at the scale

of the robot foot size. It is not possible to obtain such a map from the STOC camera range data, but it can be created by registering the sparse range data from the tilted 2D scanner or obtained from the point cloud yielded by the Xtion Pro depth camera. In the current implementation on the Messor and Messor II robots, the local grid that surrounds the robot has cells of 15×15 mm.

Its maximal dimensions on the Messor robot are 2×2 m, according to the range of the pitched scanner. These dimensions are retained for the Messor II configuration, as they roughly match the Kinect/Xtion field of view characterized by good spatial resolution of the range measurements (Park et al., 2012).

Range data from the STOC camera or the Xtion sensor working in the lower-resolution field of view allow the creating of a bigger but less precise elevation grid. The larger map size is 10×10 m, as the cameras we use can perceive objects at distances up to 5 m. Such a map can be used for coarse path planning, but is not precise enough for foothold selection, due to the limited precision of the range data at larger distances. The centers of the two maps are co-located at the origin of the robot coordinate system (Figure 5(a)). As the robot moves, both maps are translated to cover new areas. Therefore, the robot always has a precise map of the terrain in its closest neighborhood and a coarse map of the more distant areas. Note that in the experiments we store old data for visualization purposes to draw a global map of the environment from the accumulated local maps.

In Ye and Borenstein (2004), the terrain map data structure consisted of two 2D grids of the same size – one for elevation and one for certainty. Using the 3D sensors, like the stereo or depth camera, we extend this structure by dividing the space corresponding to each cell into vertically arranged bins of equal height – 50 mm in the current implementation (Figure 5(b)). Therefore, a cell in the elevation map is denoted as $h^{[i,j,n]}$, and a cell in the certainty map as $c^{[i,j,n]}$, where i, j are indices of the cells, and n is the bin index starting from the zero level.

At every time instant, new elevation measurements from the main sensor, either the STOC camera or the Xtion Pro, are registered in particular bins, depending on their $h_m = z_m + d_z + h_{ref}$ value, where d_z is the height of the sensor frame location (cf. Figure 4), and h_{ref} is the elevation at which the robot is located. On the Messor robot, the elevation measurements from the tilted laser scanner are registered in the map in the same way but using a different d_z value. The bin that contains the majority of points from the new measurements is used to calculate the new elevation value in the given 2D cell, and only this elevation is used for motion planning (Figure 5(c)). Although only one bin is selected, the measurements associated to other bins are not discarded (cf. Figure 5(a)).

The certainty value of each cell/bin counts the number of measurements from both sensors associated to this cell/bin. A single elevation measurement from a 3D sensor,

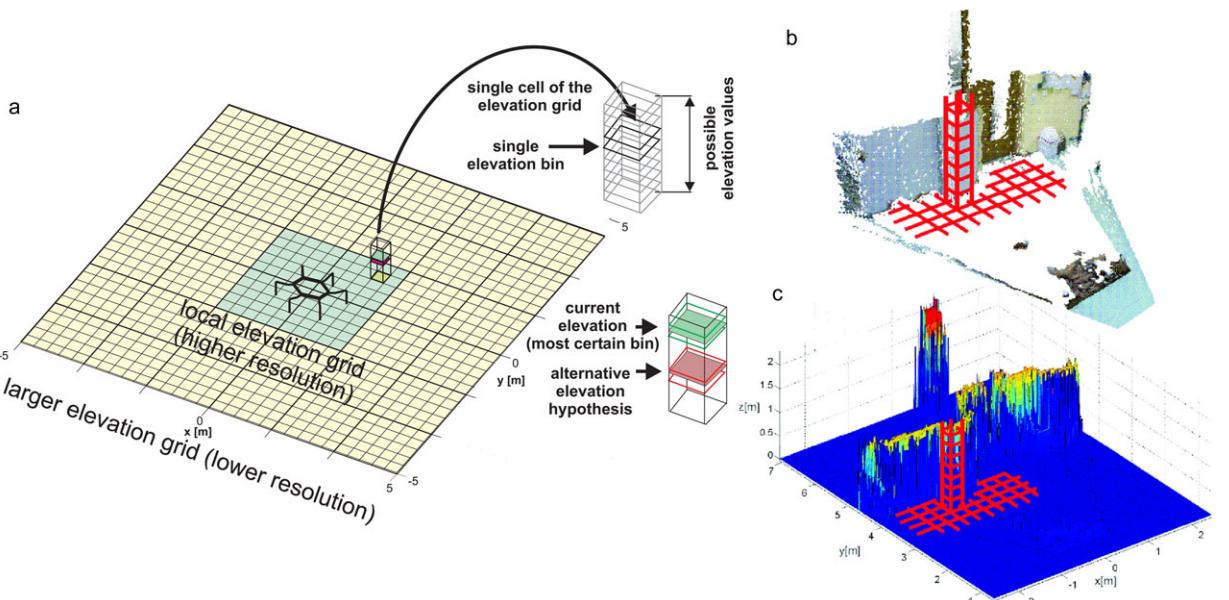


Figure 5. Concept of the elevation map with multi-bin cells (a), the close-up on the right side show the structure of the multi-bin cell, and an example of elevation estimate, the illustration of the multi-bin structure overlaid on exemplary range data from the Xtion Pro sensor, and on an exemplary elevation map is shown in the subfigures (b) and (c), respectively.

either the passive stereo or the active Xtion Pro updates the certainty according to the formula:

$$c_{(k+1)}^{[i,j,n]} = \begin{cases} c_{(k)}^{[i,j,n]} + a & \text{if } \Delta z \leq \Delta_{\max} \\ c_{(k)}^{[i,j,n]} & \text{otherwise,} \end{cases} \quad (7)$$

where a is constant increment of the certainty, Δz is the 99.7% probability interval of the elevation measurement, and Δ_{\max} is the maximum elevation uncertainty allowed for the elevation.

For the 2D laser scanner measurements, this formula is more complicated. The 2D sensor uses the robot motion to scan the terrain; thus, we apply to the range data the constraints on motion continuity, that were used in Belter & Skrzypczyński (2011b), in order to further eliminate spurious measurements:

$$c_{(k+1)}^{[i,j,n]} = \begin{cases} c_{(k)}^{[i,j,n]} + a & \text{if } (\Delta z \leq \Delta_{\max} \text{ and } |h_{m(k+1)} - h_{(k)}^{[i,j,n]}| \\ & \leq |\Delta z_{\max(k)}|) \text{ or } c_{(k)}^{[i,j,n]} = 0 \\ c_{(k)}^{[i,j,n]} & \text{otherwise,} \end{cases} \quad (8)$$

where $\Delta z_{\max(k)}$ is a prediction of the maximum change of elevation between two consecutive measurements, which is computed taking into account the maximum range measurement error and errors in the robot pose estimate. The computations of $\Delta z_{\max(k)}$ for a walking robot are the same as detailed in Belter & Skrzypczyński (2011b).

Once the most certain bin for each updated cell of the grid is determined, the elevation estimate is computed upon the collected measurements. To this end, a one-dimensional static Kalman filter is applied, which fuses all the individual h_m elevations obtained from the vertical coordinates of the p_m points accumulated in the bin, with the σ_z^2 values of the points used as weights:

$$\begin{aligned} K &= \sigma_{h(k)}^{2[i,j,n]} \left(\sigma_{h(k)}^{2[i,j,n]} + \sigma_{z(k+1)}^2 \right)^{-1}, \\ h_{(k+1)}^{[i,j,n]} &= h_{(k)}^{[i,j,n]} - K \left(h_{(k)}^{[i,j,n]} - h_{m(k+1)} \right), \\ \sigma_{h(k+1)}^{2[i,j,n]} &= (1 - K) \sigma_{h(k)}^{2[i,j,n]}, \end{aligned} \quad (9)$$

where K is the Kalman gain, σ_h^2 is the variance of the elevation stored in the given cell/bin, and $h_{(k+1)}^{[i,j,n]}$ is the updated elevation value. Note that k is iterated w.r.t. the time instances at which the range measurements were collected and that (9) can be updated asynchronously from the laser scanner (which is faster) and the stereo camera. On the Messor II robot, this is not a problem, as all the measurements are obtained at the frame rate of the Xtion Pro sensor.

The Kalman filter fuses in the statistically optimal way old elevation data (already in the map) and new elevation data, regardless of the sensor these new data are obtained from. This procedure is applied not only to the most

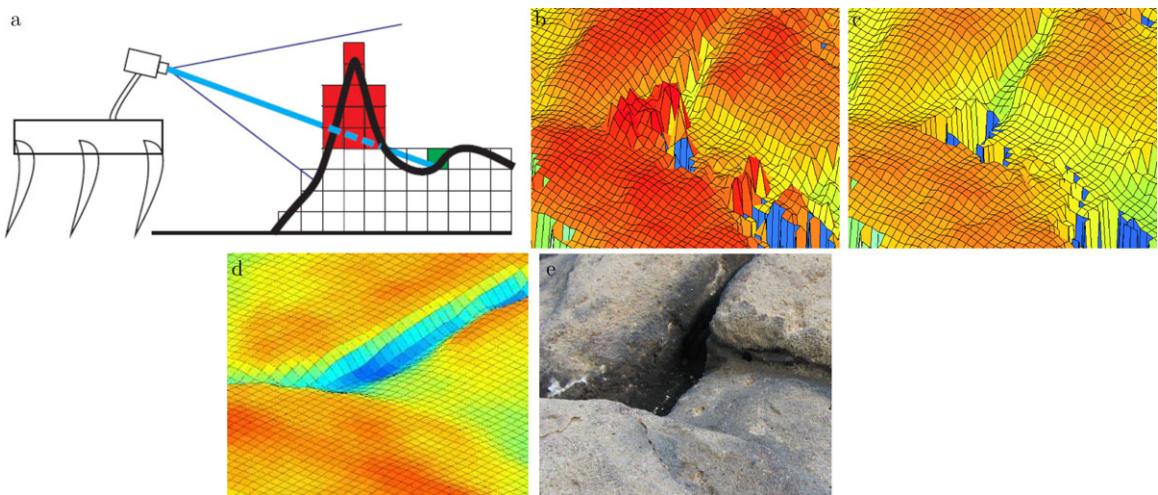


Figure 6. Application of visibility constraints: illustration of the concept, where the green cell is the one that is currently observed, and the red cells represent the elevation estimate that is probably in error (a), fragment of a map built without evaluation of visibility (b), and fragment of a map built applying the visibility constraints (c), ground truth map, (d) and a close-up photo of the real terrain (e).

certain bin, but also to other bins containing valid elevation data (i.e., the certainty of the bin is greater than zero). This way we maintain several alternative elevation hypotheses in each 2D cell of the grid, avoiding making irreversible decisions upon few range measurements.

One last step in the elevation map updating is application of the visibility constraints. If new data contain discontinuities, old but erroneous elevations can be retained in such gaps. Checking the visibility constraints helps to remove these elevations (Łabęcki, 2011). Each cell, which was not updated in the last measurement cycle is checked whether it occludes newly modified cells (Figure 6(a)). If the certainty of the cells in the gap is lesser than of the occluded cells, the measurements in the gap are removed. If the updated cell contains more than one bin with measurements, then the conflicting bin is invalidated (certainty decreased by a fixed amount), automatically making the next-best elevation hypothesis the valid elevation of that cell. An example of enforcing the visibility constraints on a real elevation map is shown in Figure 6(b) and in Figure 6(c). The simple elevation updating formula used in Ye and Borenstein (2004) and adopted in Belter & Skrzypczyński (2011b) did not allow the stored elevation values to decrease. This approach was justified for range measurements from the tilted down 2D scanner, which measured most of the cells representing flat terrain only once. However, the multisensor system needs a way to remove some wrong elevations stored already in the map from the 3D measurements, which often contain many artifacts. Another advantage of the current formulation of the elevation updating rule is that the map can accommodate changes in the terrain, providing that they are slow.

5. PERCEPTION-BASED MOTION PLANNING

5.1. Motion Planning Concept

Because of the properties of the perception system, the precision of the terrain measurements decreases with distance from the robot. The precision of the map allows only to roughly plan the path of the robot in the distant areas. However, it allows to precisely plan the full body motion of the robot on the terrain that is close to the current position of the robot.

Although the precise motion planning can be implemented efficiently by using a variant of the RRT-Connect algorithm (Belter & Skrzypczyński, 2011a), the probabilistic nature of the RRT often leads to exploring areas of the terrain that are unrelated to any reasonable path to the goal. To overcome this problem, the guided-RRT algorithm has been proposed (Belter et al., 2011). The guided-RRT planner combines our customized variant of the RRT-Connect algorithm with a grid-based A* path planner in order to keep the growing random tree close to a coarse, but reasonable path from the start to the goal.

The whole procedure (Figure 7) finds and executes a path between the start configuration (pose and posture) \mathbf{q}_{curr}^{24D} of the robot and the goal configuration \mathbf{q}_{goal}^{3D} . At first, the planning procedure finds a coarse path P_A between the current and the goal pose by using the A* algorithm. If the A* fails to yield a path, the goal is considered to be unreachable. If a path is found, a temporary goal configuration \mathbf{q}_{temp}^{3D} is created on the path P_A . The temporary goal is located at the distance d_{RRT} from \mathbf{q}_{curr}^{24D} , which at the start is equal to the current robot pose \mathbf{q}_R^{24D} . Then the RRT-based planning procedure searches for a full motion plan between

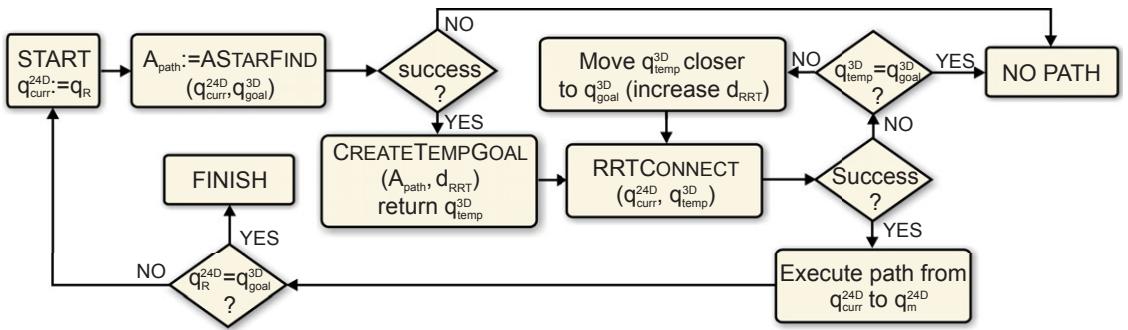


Figure 7. Block diagram of the guided-RRT algorithm.

the configurations q_{curr}^{24D} and q_{temp}^{3D} . If this search is successful, the robot executes the desired sequence of motions but only within the limits of the known local, fine-grained map. However, if the RRT-based algorithm fails to find a feasible motion sequence to q_{temp}^{3D} , the distance d_{RRT} is increased (by 0.2 m in experiments), allowing the precise motion planner to make a bigger detour from the general path found by the coarse planner.

While walking, the robot acquires new data that are stored in the local and the coarse elevation maps. After reaching the temporary goal, the path planning procedures (both A^* and RRT-based) are repeated on the updated maps. The whole motion planning procedure stops when the robot reaches the goal configuration or the RRT-based planner fails while searching for a motion plan between q_{curr}^{24D} and q_{goal}^{3D} (cf. Figure 7).

The RRT-based planning procedure can work on a single-precision and large elevation map as well as two different maps. If the considered node is located inside the local map, the robot plans its motion precisely. But if the node is outside the local map, the movement is planned by using a simplified kinematic method that roughly determines traversability between two nodes (described in subsection 5.4).

The rest of the section is organized as follows. First, we present full body motion planner, which is based on the RRT-Connect framework. Then, we present software modules, which support precise motion planning and allow the reduction of dimensionality of the problem. We describe foothold selection method, posture optimization strategy and feet trajectory planner, and methods that check whether the planned path is secure (self-collisions free). Finally, the simplified cost computation for A^* planner is presented.

5.2. Precise Motion Planning

To find a precise sequence of movements for the legged robot, an algorithm working in a continuous search space should be used, thus graph search methods like A^* are unacceptable. We chose an RRT-based planner because of

its ability to quickly explore the high-dimensional search space. The precise motion planner is based on the RRT-Connect algorithm (Kuffner Jr. & LaValle, 2000), which is used here as a meta-algorithm, invoking several lower-level planning modules. The particular subprocedures of this meta-algorithm are explained further in the paper.

Like RRT-Connect the RRT_MOTION_PLANNING procedure (Algorithm 1) creates two random trees. Each node in a tree represents a kinematic configuration of the robot. A child node is connected to the parent node if a transition between these two configurations is possible. The root configuration of the first tree T_c is located in the current configuration q_c . The root configuration q_t of the second tree T_t defines a temporary goal state of the robot. For the T_c tree, the motion is planned forward, whereas the direction in the T_t tree is inverted.

Algorithm 1 RRT_Motion_Planning (q_c, q_g)

Data: initial q_c and goal q_g poses of the robot

Result: precise motion path PATH(T_c, T_t)

$T_c.Init(q_c); T_t.Init(q_t); k:=1;$

while ($k < K$) **do**

$q_{rand} := \text{RANDOM_CONFIG};$

if not ($q_{new} := T_c.\text{EXTEND}(q_{rand})$) = Trapped **then**

if $T_t.\text{EXTEND}(q_{new})$ = Reached **then**

return PATH(T_c, T_t);

end

end

 SWAP(T_c, T_t);

$k := k + 1;$

end

At first, the procedure RANDOM_CONFIG randomly selects a pose of the robot q_{rand} (only the x and y coordinates) on the given elevation map (Belter, 2013). Then, the EXTEND operation extends the tree and tries to build new branch

in the direction of q_{rand} . If it is possible, and the new node q_{new} is created, the algorithm extends the second tree in the direction of q_{new} node. In the next iteration of the algorithm, the order of the trees is swapped (SWAP procedure). As a result, the T_t tree is extended at the beginning of the next iteration. The maximal number of iterations is defined by K .

When the two trees are connected and the EXTEND procedure returns "Reached" value, the algorithm finds the shortest path between q_c and q_g . The q_{new} configuration is common for the two trees. The PATH procedure finds the path from q_{new} to the root nodes in both trees. The sequence of the nodes for T_t tree should be inverted. Finally, the sequence of configurations which allows movement from the initial to the goal configuration is obtained.

The EXTEND procedure (Figure 8) plays a key role in the RRT-based planner for a legged robot. It checks whether a transition between two given configurations of the robot is possible and extends the tree. First, the procedure determines the maximal step length in the desired direction (d_{STEP}). The maximal step length is found using kinematic and collision constraints. To this end, the procedure finds the existing node q_{near} , which is the closest one to the new, random position q_{rand} . To determine the maximal length of the step, we create initial posture of the robot q_{simp} . The procedure assumes initial horizontal orientation of the robot's body. The distance to the ground is set to guarantee secure clearance between robot's body and the ground (in experiments, the clearance is set to 0.1 m). The subprocedure, which plans the path for the next robot step, is executed five times for various step length $k \cdot d_{\text{STEP}}$ ($k \in \{0.2; 0.4; \dots; 1.0\}$). Then, a new configuration is created by using the module, which selects footholds and creates a posture (Belter & Skrzypczyński, 2011b, 2012). Next, the procedure plans the path of feet during swing phase. The path of the body is straight between two neighboring nodes. Then, the planner checks whether the desired sequence of the robot's postures is achievable (lack of collisions, acceptable footholds inside the workspaces) and safe (robot maintains the static stability). Finally, the robot chooses the longest possible step.

5.3. Main Modules of the Precise Motion Planner

5.3.1. Foothold Selection Method

At a given posture of the body, the robot has to find appropriate footholds. The algorithm proposed in Belter et al. (2010) is used. The detailed description of the method is presented in Belter & Skrzypczyński (2011b). The aim of this algorithm is to find an analytical relation between some geometrical characteristics of the local shape of the terrain, and the predicted slippage of the robot's feet. The predicted slippage is a slippage that is expected when the robot chooses the considered point of the elevation map. As shown in Belter & Skrzypczyński (2011b), the robot learns how to determine the footholds from the experience. Then it exerts the

obtained experience to predict the slippage and minimize it by choosing proper points from the map. The result of the algorithm on an example map is presented in Figure 9(a). The module allows the avoidance of risky poses of the feet (on edges or on slopes).

5.3.2. Posture Optimization Strategy

The module responsible for maintaining an acceptable posture creates the robot's configuration at the location on the elevation map (x_R , y_R). The horizontal position of the robots body is provided by RRT planner. It computes the required inclination of the robot's trunk (θ_R , φ_R) and its elevation z_R above the terrain.

The optimization procedure that solves the problem searches for the vector $\mathbf{p}_R = [\theta_R, \varphi_R, z_R]^T$, which determines the posture of the robot with maximal kinematic margin of the robot d_{KM} . The kinematic margin d_{KM}^i of the i -th leg is defined as the Euclidean distance from the current position of the foot to the boundary of the reachable area of the leg (Kumar et al., 2002). To compute kinematic margin d_{KM} of the robot, the distance from current positions of the feet to the boundaries of the legs workspace are computed. The smallest distance ($\min(d_{\text{KM}}^{i=1}, \dots, d_{\text{KM}}^{i=6})$) determines the d_{KM} value (Belter & Skrzypczyński, 2012). The formal description of the optimization procedure is as follows:

$$\arg \max_{\mathbf{p}_R} \{d_{\text{KM}}(\mathbf{q}(\mathbf{p}_R))\}, \mathbf{q}(\mathbf{p}_R) \in C_{\text{free}}, \mathbf{q}(\mathbf{p}_R) \in C_{\text{stab}}, \quad (10)$$

where $\mathbf{q}(\mathbf{p}_R)$ – full state of the robot: platform configuration \mathbf{p}_R and position of feet determined using foothold selection algorithm,

$d_{\text{KM}}(\mathbf{q}(\mathbf{p}_R))$ – kinematic margin of the robot for the given state of the robot $\mathbf{q}(\mathbf{p}_R)$,

C_{free} – configurations of the robot which are collision-free (lack of collisions with the ground and between parts of the robot) and inside of the workspace of the robot,

C_{stab} – statically stable configurations of the robot. The static stability of the new posture is checked using a simple but fast procedure implementing the stability criterion defined by (McGhee & Iswandhi, 1979).

The problem defined here can't be solved defectively using gradient method (the gradient of the goal function can't be computed) thus to find the optimal posture of the robot we use the particle swarm optimization (PSO) algorithm (Belter & Skrzypczyński, 2012; Kennedy & Eberhart, 1995). The PSO algorithm creates a set of particles in the solution space. Each particle represents solution of the problem. During single iteration of the algorithm, a particle moves in the direction combined according to the best global solution and the best position in the history of the considered particle (Kennedy & Eberhart, 1995).

The computation of the kinematic margin is performed hundreds of times during a single optimization run. To speed up the computation, we approximate the

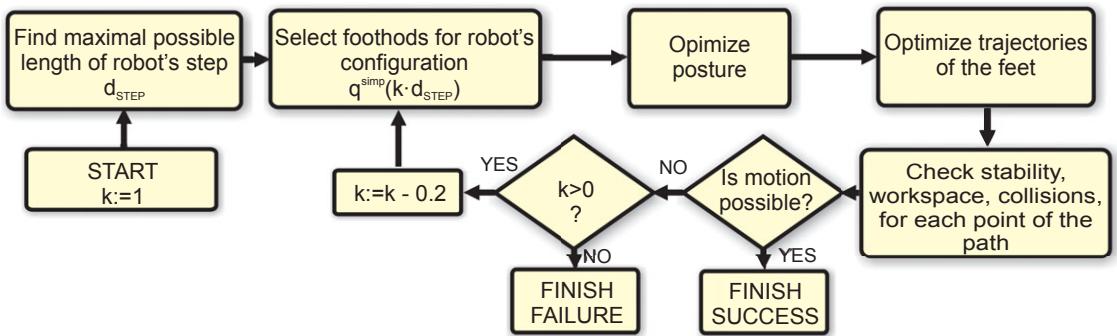


Figure 8. Block diagram of the EXTEND procedure in the RRT-based motion planner.

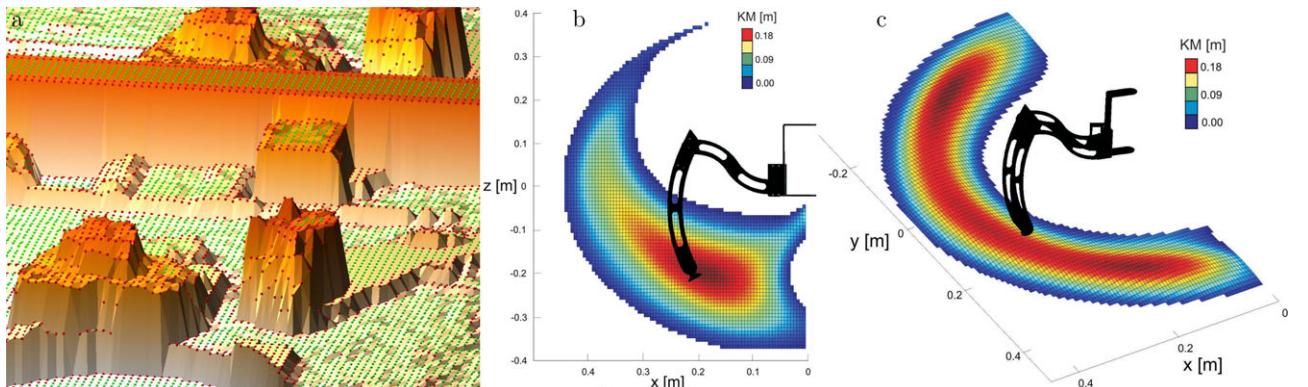


Figure 9. Visualization of exemplary results provided by two modules of the motion planner: the foothold selection module (a), and the posture optimization module. In (a) the results are color-coded: green – acceptable footholds, red – unacceptable footholds. Subfigures (b) and (c) show respectively the vertical and horizontal cross-sections of the workspace approximation of the robots legs – colors represent the kinematic margin, i.e., the distance to the border of the workspace.

legs workspace d_{KM}^i with mixture of Gaussian (Belter & Skrzypczyński, 2012). The mixture of Gaussian finds the regression of the function:

$$d_{KM}^i = f(x_f^i, y_f^i, z_f^i), \quad (11)$$

where $[x_f^i, y_f^i, z_f^i]$ is the position of i -th robot's foot. The approximation of the legs workspace is presented in Figure 9(b) and (c). The approximated function (workspace) returns kinematic margin of the leg for the given foot position.

5.3.3. Feet Trajectories Planner

In the transition between two configurations, the path of the robot's trunk is determined as a straight line. However, the motion planner also has to determine whether the robot is able to move its legs in such a way that the goal state is reached, for example, by moving the legs above an obstacle. Therefore, a fast trajectory planner for the feet of the robot is implemented, which takes into account the shape of the terrain as modeled by the detailed elevation map.

The straight trajectory above the obstacles might be impossible to execute because the foot pose above the obstacle causes collisions between the robot's parts or the pose is outside the workspace of the robot. Also the length of the path planned using simplified method might be far from optimal. Moreover, the execution of the planned path looks unnatural because the robot lifts its feet very high. We solve this problem by using optimization of feet positions during swing phase. We run the optimization procedure whenever the kinematic margin of the considered foot is below the threshold.

The goal of the optimization (suboptimization) is to find the position of the i -th foot during swing phase $\mathbf{p}_f^i = [x_f^i, y_f^i, z_f^i]^T$ (expressed in the global coordinate system O), which satisfies the requirements for the value of the i -th leg kinematic margin d_{KM}^i :

$$d_{KM}^i(\mathbf{p}_f^i) > d_{KM}^{\text{safe}}, \mathbf{p}_f^i \in C_{\text{free}}, \quad (12)$$

where C_{free} are leg's configurations, which are collision-free and inside of the workspace of the robot.

We decided to use suboptimization instead of full optimization because looking for leg-end position, which satisfies safety requirements, is sufficient. The suboptimization stops searching when the kinematic margin is bigger than $d_{\text{KM}}^{\text{safe}} = 8$ cm. This approach also speeds up the optimization procedure. If necessary, the procedure is run for each foot and all points of the initial path during swing phase. Again we use the approximation of the leg's workspace (Figure 9(b) and (c)). To find the optimal foot position, the method of steepest descent is used (Kelley, 1999).

The procedure for planning trajectories of the feet also sets the sequence of swing and stance phases for the legs, which defines the gait. Currently, the motion planner applies wave, tripod gait, and free gait (Belter, 2012). However, in the experiments, we use tripod gait only. We prefer this gait because it is the fastest statically stable gait for six-legged walking robot.

The planner checks whether each foot position is located inside the workspace of the respective leg. Moreover, legs of the robot can collide if a selected foot position is inside of the workspace of the other legs. To detect such situations, the planner is equipped with a module for collision detection, which uses a CAD model of the robot. Collision detector performs the triangle-triangle intersection test for oriented bounding box (OBB) (see Belter & Skrzypczyński (2011a) for details).

5.4. Coarse Path Planning

The A* algorithm performs coarse path planning on the map of distant areas obtained from the lower-quality 3D data. This representation is sufficient for coarse path planning, which allows to obtain a general direction of further motion, that is, to guide the RRT-based planner. Moreover, the lower-resolution grid allows A* to run in real time.

The procedure ASTARFIND (cf. Figure 7) performs path planning by using the larger, lower-resolution map. The A* algorithm is chosen for this purpose. Nodes are located at the center of the grid cells, so the distance between two nodes is 0.1 m. It is sufficient for coarse path planning. For this resolution, the A* is still fast enough for these purposes, and it allows to obtain a general direction of the further movement, that is, to guide the RRT motion planning. The A* algorithm minimizes the function:

$$f(\mathbf{q}^{2D}) = g(\mathbf{q}^{2D}) + h(\mathbf{q}^{2D}), \quad (13)$$

where $h(\mathbf{q}^{2D})$ is the predicted cost of the path between the position \mathbf{q}^{2D} and the goal node. Euclidean distance is used to define the heuristic $h(\mathbf{q}^{2D})$. The function $g(\mathbf{q}^{2D})$ is the cost of the path between the initial node and the node \mathbf{q}^{2D} . To compute $g(\mathbf{q}^{2D})$, a cost of transition between neighboring nodes is defined. To do so, two coefficients c_1 and c_2 that depend on the terrain characteristics are used, and the c_3 coefficient that depends on the instantaneous kinematic configuration of the walking robot.

The first coefficient c_1 defines the Euclidean distance between neighboring nodes. All coefficients should have values in the same range. It allows to compare the values of the coefficients so the value of c_1 is divided by 0.25 to normalize it into the range between 0 and 1. The bigger the value of the coefficient c_1 , the bigger the cost of transition between the nodes.

The spherical variance ω (Sanctis et al., 2009) is used to define the second coefficient c_2 . The grid map is converted to a triangle mesh by using the considered point and its neighbors (the number of triangles n is 8). For each triangle on the surface, the vector $\vec{\mathbf{N}}_i = (x_i, y_i, z_i)$ normal to the triangle is computed. Then, the module R of the vector set $\vec{\mathbf{N}}_i$ is computed as follows:

$$R = \sqrt{\left(\sum_{i=0}^n x_i\right)^2 + \left(\sum_{i=0}^n y_i\right)^2 + \left(\sum_{i=0}^n z_i\right)^2}. \quad (14)$$

The module R is divided by the number of vectors n to normalize the values into the range between 0 and 1. Finally, the spherical variance is computed as follows:

$$c_2 = \omega = 1 - \frac{R}{n}. \quad (15)$$

The value of c_2 defines the "roughness" of the terrain. The higher the value, the bigger difficulties the robot has in finding appropriate footholds.

To compute the final coefficient c_{final} (16), the kinematic capabilities of the robot to traverse between two nodes are also considered (c_3 coefficient). The computation is simplified. The roll, pitch, and yaw angles of the trunk are set to zero and the robot places its feet on the nominal footholds. Finally, if the goal state is acceptable and a transition between the initial and the goal state is possible (c_3 is set to 1), the coefficient c_{final} is computed as the mean value of the coefficients c_1 and c_2 . If the transition is not achievable (c_3 is set to 0), the final cost c_{final} is set to infinity.

$$c_{\text{final}} = \begin{cases} \frac{c_1+c_2}{2} & \text{if transition is possible } (c_3 \text{ is } 1), \\ \infty & \text{if transition is not possible } (c_3 \text{ is } 0). \end{cases} \quad (16)$$

6. ADAPTATION OF THE COARSE PATH PLANNER

The heuristic used to set the $h(\mathbf{q}^{2D})$ value, which was presented in the previous section works efficiently enough to plan the motion of the robot on rough terrain. When the robot approaches the obstacle, it collects new and more detailed data about the terrain. Then, the RRT-based planner can find a shorter path despite the fact that A* guided it to a wrong direction. However, the A* planner with the presented heuristic for traversability cost estimation is overly conservative. In many cases, it avoids obstacles that are in fact traversable to the walking robot. Thus, the aim of the novel prediction-based approach presented here is to reduce the number of wrong assessments.

6.1. Prediction with RRT-based exploration

The idea of predicting the traversability assessment is to use the RRT-based planner to obtain reference values of the traversability cost, and then to associate these values to some observable characteristics of the terrain. The RRT-based planner takes into account the detailed geometric characteristics of the terrain and evaluates the possibility of motion with regard to the robot kinematics and various constraints, such as the static stability, and possible collisions.

In our approach, we use maps from previous experiments on rough terrain to create a reference map for prediction. Then, we use the RRT algorithm to plan the motion of the robot. Traversability between particular cells of the map is determined by selecting proper footholds, determining feet trajectories, and taking into account all constraints. Simultaneously, the prediction procedure decreases the resolution of the original elevation map to achieve the level of details typical to the long-range map used by the A* planner. The coefficients c_1 , c_2 , and c_3 are computed and stored as an input to the regression algorithm. The output is the binary information p_t from the RRT planner. The value of p_t is 1 if the transition between the given cells is possible (according to RRT planner, which uses high-resolution grid map), and 0 if the transition is not possible.

6.2. Traversability Assessment

We decided to use a probabilistic framework (kernel density estimation – KDE) to assess the traversability. The objective is to find the probability density $p(p_t | c_1, c_2, c_3)$, which means that we are looking for the probability that a transition between two nodes is possible given the values of coefficients $\mathbf{c} = [c_1, c_2, c_3]^T$ describing the terrain traversability between these cells. To find the conditional probability density, we estimate the joint density $p(p_t, c_1, c_2, c_3)$, and marginal density $p(c_1, c_2, c_3)$ using (17), and (18) (Hansen, 2004):

$$p(p_t, c_1, c_2, c_3) = \frac{1}{n} \sum_{i=1}^n K_{h_2}(\mathbf{c} - \mathbf{c}_i) K_{h_1}(p_t - p_{ti}), \quad (17)$$

$$p(c_1, c_2, c_3) = \frac{1}{n} \sum_{i=1}^n K_{h_2}(\mathbf{c} - \mathbf{c}_i), \quad (18)$$

where \mathbf{c}_i and p_{ti} are the observed (training) samples of \mathbf{c} and p_t , respectively. K_{h_1} and K_{h_2} are Gaussian kernels, and n is the number of samples. The conditional density function is computed as follows:

$$p(p_t | \mathbf{c}) = \frac{p(p_t, \mathbf{c})}{p(\mathbf{c})}. \quad (19)$$

To find the maximal probability and the value of p_t which corresponds to the given value of \mathbf{c} , we solve the optimization problem:

$$p_t = \arg \max_{p_t} p(p_t | \mathbf{c}) \quad (20)$$

by using the mean-shift algorithm (Fukunaga & Hostetler, 1975). The output p_t from KDE is a continuous value in the range 0 to 1. We discretize the output. The output value in the range 0 to 0.5 is considered as nontraversable, bigger than 0.5 as traversable.

7. RESULTS

7.1. Determining Feasibility of the Planning Approach

To determine how the proposed approach to motion planning works for various types and configurations of the environment, and to make the results independent from the unavoidable errors in perception, mapping, and self-localization, we demonstrate some results of simulations. The realistic simulator of the Messor robot is based on the open dynamics engine (Smith, 2014), and has been already successfully used to learn efficient gaits of a hexapod (Belter & Skrzypczyński, 2010), and to develop the foothold selection policy for Messor (Belter & Skrzypczyński, 2011b), which is also applied on Messor II. For the motion planning simulations, a terrain map has been prepared. This map includes a model of a rocky terrain mockup acquired using a laser scanner moved precisely by an industrial robot arm. In addition, it includes some synthetic objects, such as a nontraversable hill, four nontraversable walls, and one traversable wall. The notion of nontraversability is here related to the kinematic constraints of the walking robot, mainly to the length of its legs.

An example path obtained by running the guided-RRT algorithm on this map is shown in Figure 10. The algorithm successfully found a path to the goal position. The robot avoided nontraversable obstacles and also found a way to deal with the traversable obstacles (mockup and traversable wall) (Figure 10). The problem is solved in two stages – coarse planning, which gives only a general path toward the goal, and precise motion planning, which yields a full plan for the robot body and feet. If the goal is far from the current location, the robot can only find a direction of movement, follow this path, and locally solve the problem of selecting footholds, crossing small obstacles, or going around bigger ones.

Another simulation presented in Figure 10 shows some interesting properties of the algorithm. The region $R1$ was assessed as traversable by the A* planner. When the robot reached this area, it turned out to be too difficult to find an appropriate path. The RRT algorithm found a new path that goes round the nontraversable area. The second interesting situation is related to the region $R2$. The A* assessed it as

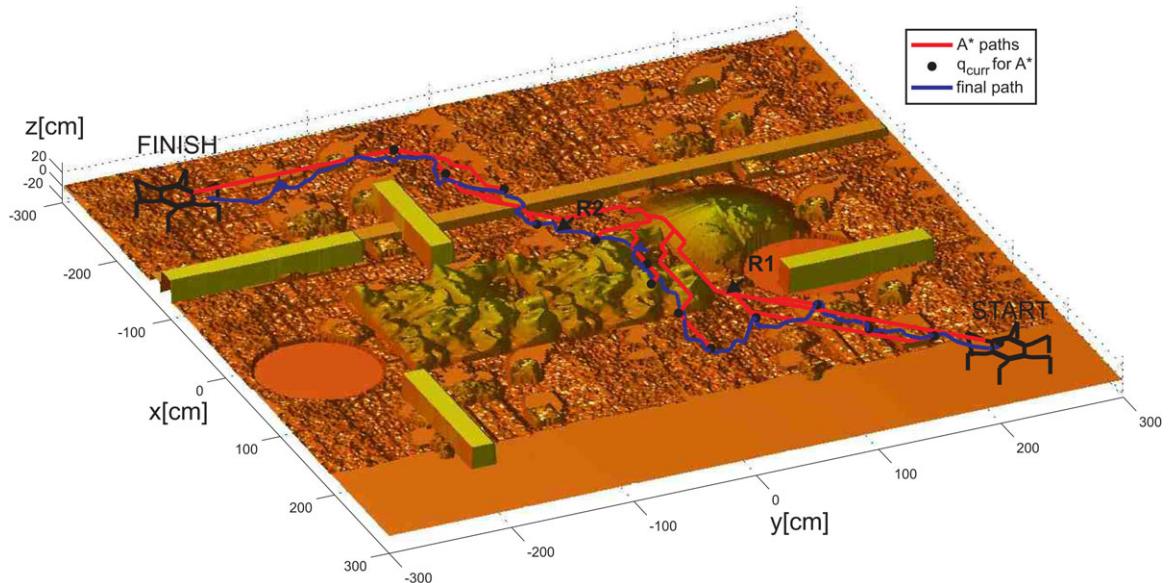


Figure 10. Exemplary final path obtained by the guided-RRT algorithm in a simulation on a map of rough terrain with various obstacles.

nontraversable. The RRT planner found a path across this region, thus shortening the initial path.

Other properties of the guided-RRT are demonstrated in Figure 11. The right image presents a path found by using only the RRT-Connect algorithm. The left image presents all RRT trees created by the presented algorithm. The original RRT algorithm searches over the whole map, which takes a lot of time. Also the final path is far from optimal. The RRT planning in the proposed approach is focused on exploring the areas around the optimal path given by the A* planner thus the proposed method returns shorter path. The planning using guided-RRT is a few times faster in this experiment not only because the total length of the RRT path is smaller but also because the pure RRT algorithm explores areas that are not traversable by the robot and fails to extend the tree two to three times more frequently than the guided-RRT.

To show the efficiency of the algorithm, a series of three simulations were conducted. The environment, initial, and goal position of the robot are the same in the series of simulations. Because the RRT is the random-based algorithm, we want to check the convergence of the whole method. The obtained paths are shown in Figure 12(a). Although the RRT sampling is random, the results are similar. The robot avoids the same nontraversable areas. Only the local paths are slightly different. Another simulation was run on the synthetic map shown in Figure 12(b). This map includes a T-shaped obstacle. Such a map is used to check whether the proposed algorithm can first lead the robot round the obstacle (the distance to the goal increases),

to finally reach the goal. The obtained path is shown in Figure 12(b).

To find the relation between the guided-RRT planning time and roughness of the terrain, we performed a series of simulated experiments on a real terrain map obtained with the Messor II robot. The results are presented in Figure 13(a). For the simulations, we modify the elevation values of this map using a roughness scaling factor k_r in the range from 0 to 1. Thus, the elevation of each cell in the prepared map is computed as $h_{\text{new}}^{[i,j]} = k_r \cdot h^{[i,j]}$. Two example maps prepared for the simulations are presented in Figure 13(b) and (c). When the roughness scaling factor is 0, the terrain is completely flat. When the scaling factor is 1, the height of each cell in the map remains unchanged, which means that the RRT-based planner has to negotiate obstacles as high as 25 cm (the large wall in Figure 13(c) is avoided by the coarse planner). The motion planning horizon is 2 m. We run these experiments on a PC with 2.8 GHz Intel Core i7. For each scaled map, we run a series of 10 planning experiments. Then, we compute the average planning time, its standard deviation, and the success ratio. The planning is considered successful if the RRT EXTEND operation (cf. Figure 8) is invoked no more than 300 times, and the whole guided-RRT procedure finds a feasible motion plan in less than 300 s. For flat terrain, the average planning time is 13.4 s. Note that the planning method runs in a single thread and can be further optimized to decrease the execution time. The planning time increases gradually as the roughness of the terrain increases. The success ratio is 100% for the scaling factors up to 0.8 and then drops as the roughness becomes

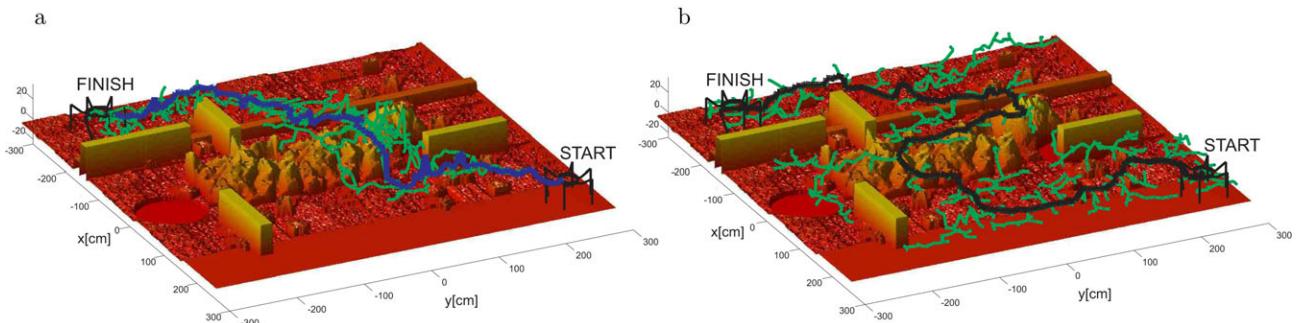


Figure 11. The final path obtained by the A* guided-RRT algorithm (a), and the path obtained by our implementation of the RRT-Connect algorithm for the same planning task (b).

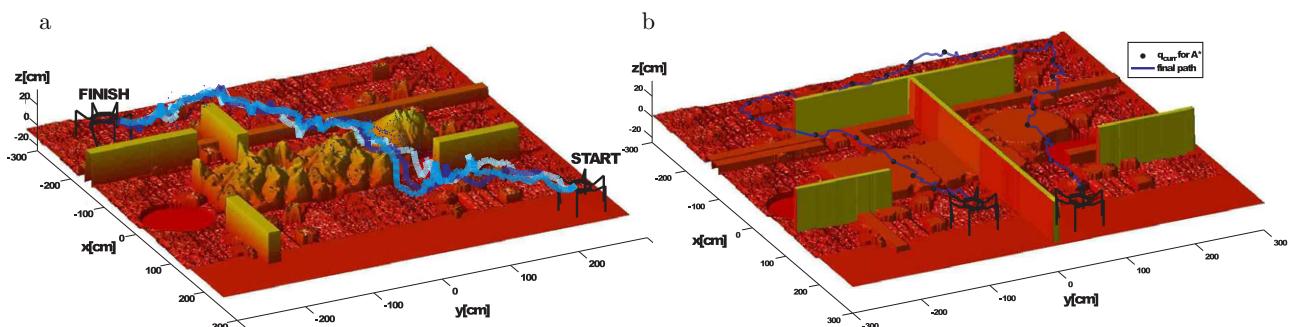


Figure 12. A series of path-guided RRT planning experiments (a) and results obtained on a synthetic map with a T-shaped obstacle (b).

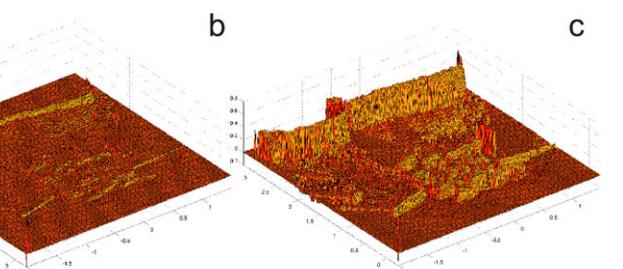
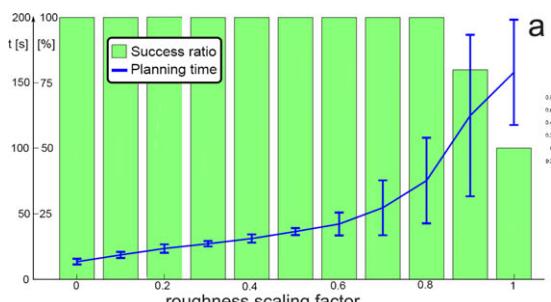


Figure 13. Execution time of the guided-RRT planner as a function of the terrain roughness scaling (a), and sample terrain maps for different scaling factors: k_r = 0.1 (b) and k_r = 1.0 (c).

hardly manageable taking into account the length of the robot's legs.

Although the simulations demonstrate that the proposed guided-RRT approach is efficient in searching for a motion plan even in relatively complicated environments, these results have been obtained assuming availability of a perfect terrain map and no localization errors. Therefore, to verify the concepts related to perception and mapping in the presented system, we performed small-scale indoor and outdoor experiments, involving the Messor robot and a mockup of a rocky terrain or a real rubble-covered ter-

rain with some vegetation. In these experiments, the whole system did not work in real time. The mapping procedure, which used data from the URG-04LX laser scanner and the stereo camera, worked off-line. The robot with the two sensors was moved along a long tracklike meter to avoid larger localization errors. Thus, the motion sequences were planned for the robot but actually not executed by the physical robot. Nevertheless, these experiments enabled us to verify the properties of the motion planner on terrain maps acquired from real multisensory data.

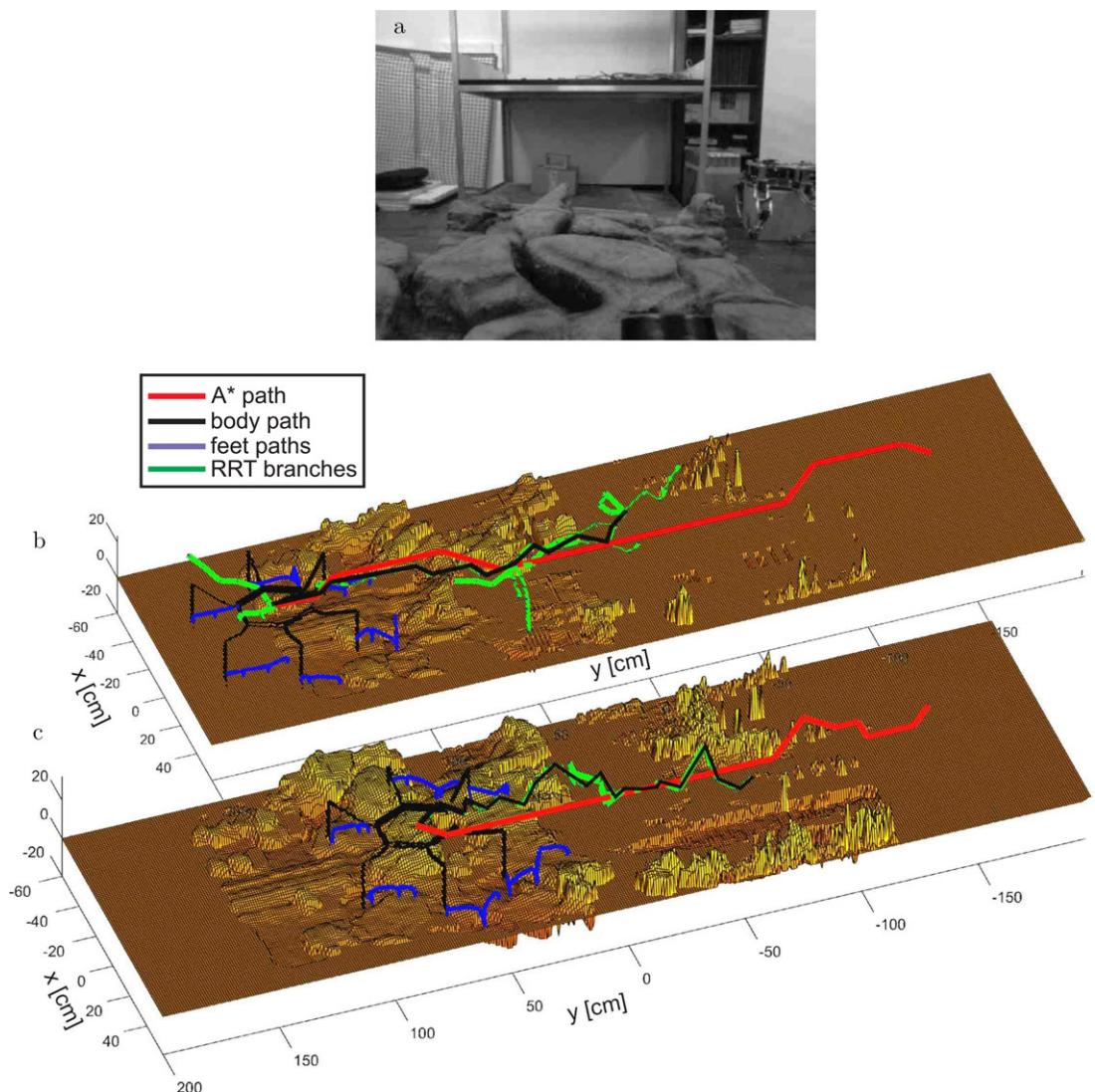


Figure 14. Results of testing the guided-RRT algorithm with the elevation map of a rocky terrain mockup obtained from perception: experimental setup (a), and two snapshots of the map and paths (see text for the explanation of the meaning of colors) during the experiment (b,c).

The indoor experimental setup is shown in Figure 14(a). Figures 14(b) and (c) visualize two moments during an indoor experiment: when the robot is about 40 cm from the start point (b), and when it is about 1 m from the start (c). The general path provided by the coarse path planner is shown in red, the RRT tree is shown in green, while the final path is black. The thin blue lines depict the local trajectories planned for the robot's feet. One can see that the path found by the RRT-based motion planner differs slightly from the coarse path given by the A* planner (Figure 14(c)). It is also visible that both the coarse path and the motion plan get updated as the robot acquires new in-

formation about the terrain – this could not be observed in simulations.

Figure 15 shows a similar situation during an outdoor experiment, illustrating the ability of the Messor's onboard perception system to handle natural scenes. The experimental setup is shown in Figure 15(a). The distance between the configurations in Figure 15(b), and Figure 15(c), and Figure 15(d) are 50 cm. During execution of motion, the robot gradually acquires new information about the terrain. The shapes of the obstacles, for example, stones and boxes become more distinguishable. The number of occluded (empty) areas in the map also decreases. At the

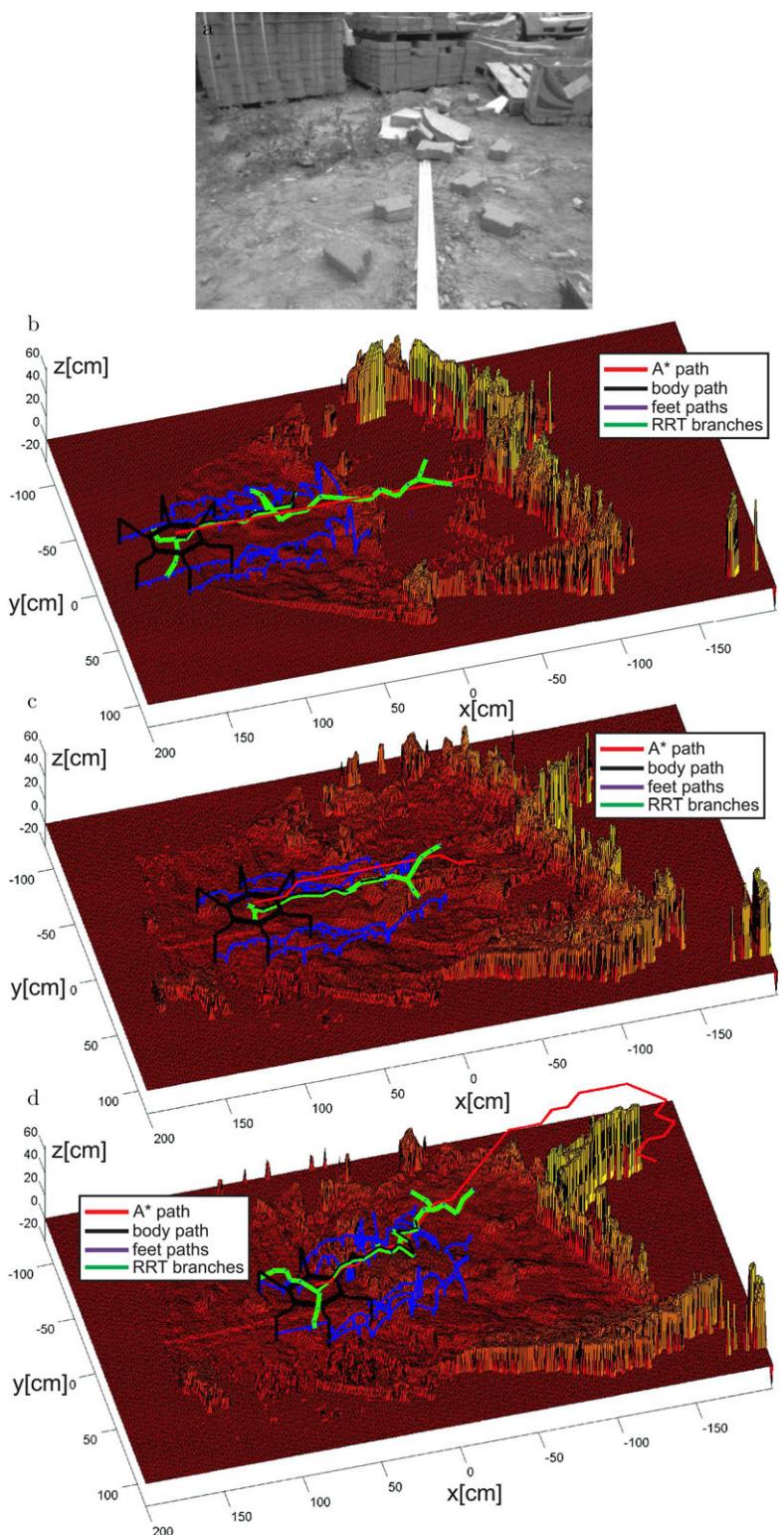


Figure 15. Results of testing the guided-RRT algorithm outdoors in a rubble-covered area: view of the experimental site (a), and three snapshots of the map and paths (see text for the explanation of the meaning of colors) during the experiment (b,c,d).

last iteration of the motion planner (Figure 15(d)), the goal position was moved farther from the robot. In this situation, the robot found a new path around the box. This illustrates the ability of our approach to cope with semidynamic situations, for example, when during a USAR mission, the human operator has to change the goal location while the robot is moving.

Note that the map in Figure 15(c) is created using the stereo camera only. The map presented in Figure 15(b) was created when the robot executed the first planned path. When the robot moves, the map is updated using data from the stereo camera and the Hokuyo URG-04LX laser scanner. The field of view of the URG-04LX sensor is 240°. Thus, despite of the fact that it is mounted at the front of the robot, the scanner can measure objects that are 1.4 m behind it. Because in our configuration, the scanner is tilted down, and the range is smaller and limited to high obstacles. Still, during the experiment, the scanner was able to measure objectsthat were sideways of the robot.

7.2. Verification of the Adaptation Mechanisms

The initial verification of the motion planning method presented in the previous subsection enabled us to realize that in many situations the A* planner yields a path that unnecessarily avoids areas of the map that are in fact traversable to the walking robot. This problem is attributed to the fixed traversability coefficients used by the planner. The computation of the c_{final} coefficient (16) is a heuristic that works for most of the terrains but fails for more demanding areas, where the proper balance between the “roughness” of the terrain and the Euclidean distance to the next configuration is harder to determine. Our approach to circumvent this problem is the regression-based adaptation of the traversability described in Section 6.

To demonstrate quantitatively the gains due to this adaptation mechanism, we collected data for the regression on the map depicted in Figure 16. In Figure 16(a), we presented the RRT tree and the assessment results before prediction (using the default coefficient c_3 (16)). We use the following notation: TP (True Positive) – the terrain is traversable and the robot made a correct decision; TN (True Negative) – the terrain is nontraversable and the robot made a correct decision; FP (False Positive) – the terrain is nontraversable, and the robot made an incorrect decision. FN (False Negative) – the terrain is traversable, and the robot made an incorrect decision.

With the baseline approach, the percentage of correct assessments was 78.0% (TP+TN, see Table I). Using the baseline method, the A* algorithm made 20.3% of wrong assessments about transitions that are traversable but were assessed as nontraversable and only 1.7% wrong assessments about transitions, which are not traversable (and were assessed as traversable). When the prediction-based approach is used, the assessment is more balanced. The

Table I. Efficiency of traversability assessment.

		Correct [%]	FP [%]	FN [%]
Prediction	before prediction	78.0	1.7	20.3
	after prediction	86.6	7.8	5.6
Test set 1	before prediction	53.2	5.9	40.9
	after prediction	63.8	15.2	21.1
Test set 2	before prediction	51.8	3.1	45.1
	after prediction	70.1	4.7	25.2

number of proper assessments increases to 86.6% (note that 100% efficiency is not possible because the traversability assessment module works on the low-resolution map from long-range perception, and therefore does not have detailed information about the terrain). We also performed verification simulation on the map, which was used for prediction (Figure 17). The number of correct assessments before (Figure 17(a)) and after prediction (Figure 17(b)) is 53.2% and 63.8%, respectively.

7.3. Experimental Verification of the Fully Integrated System

To show the feasibility of the presented approach to perception-based motion planning in the context of USAR applications we performed a series of experiments with the walking robots Messor and Messor II. In these experiments, the fully integrated system has been verified, including the coarse path planner, the motion planner (i.e., the components of the guided-RRT solution), the adaptive traversability assessment procedure, the online terrain mapping module, and the self-localization based on PTAM or the legged odometry and AHRS. In fact, many more procedures were involved in this system “under the hood,” as the precise motion planner invokes a number of quite sophisticated software modules in order to check the constraints related to the footholds, kinematics, feet trajectories, and collisions.

We started with an experiment performed indoors using the older Messor robot with the multisensor perception system. The experimental setup is shown in Figure 18. The goal position is about 3 m in front of the robot (ratio between the distance covered by the robot and its length was about 8). The Messor robot should avoid the obstacles on its path (Pioneer robot and some artificial rocks), and it should create an elevation map of the environment. Figure 19 presents the following sequences of motion planning and motion execution procedures. Each subfigure depicts a path found by the A* planner, the whole RRT tree, and the final path executed by the robot. For the sake of clarity, the subfigures do not show feet paths.

In Figure 19, one can also observe the representations of the real objects: legs of the table (arrow 1 in Figure 18(b) and in Figure 19(c)), Pioneer robot (arrow 2 in Figure 18(a) and

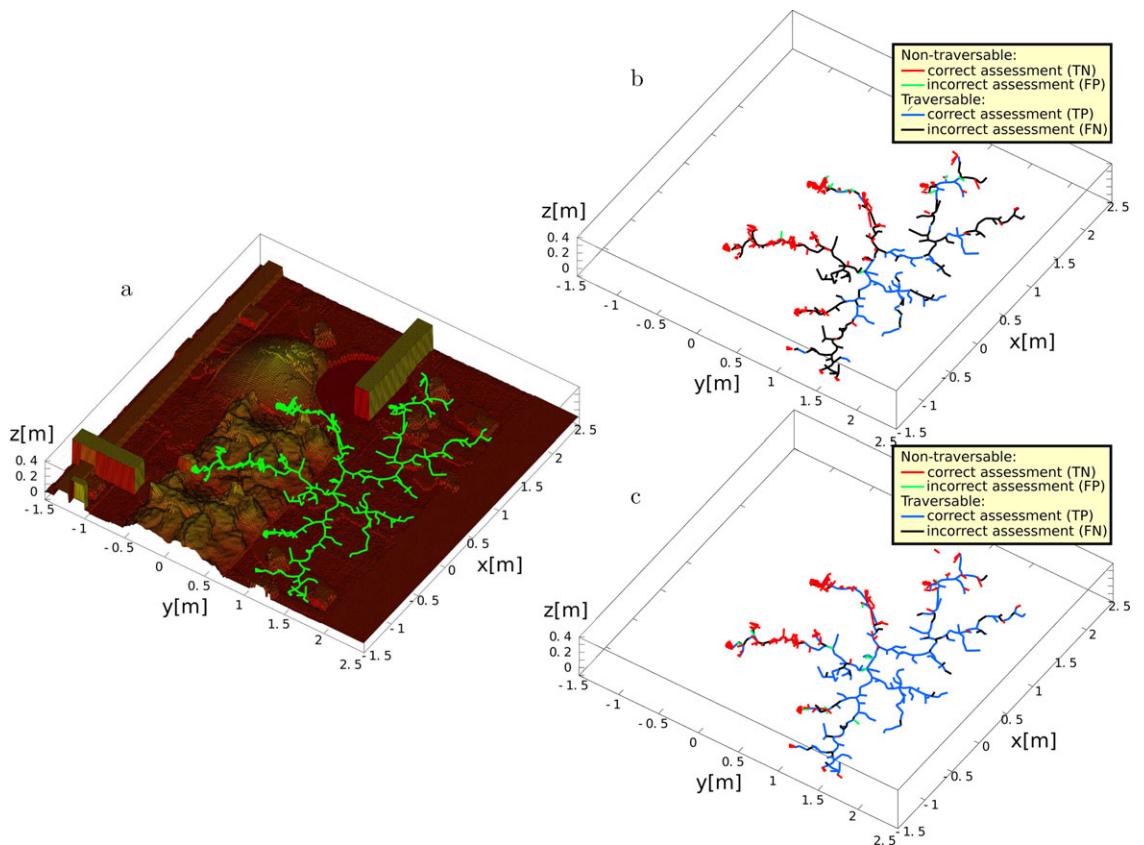


Figure 16. Traversability assessment before (b) and after prediction (c) – learning set on the rough terrain (a). Each edge represents a single step of the robot. The reference traversability assessment is provided by the RRT-based planner. The assessment is made using hand-tuned heuristic (b) and KDE predictor (c). The colors of the edges represent the result of the assessment.

Figure 19(f)) and artificial rocks (arrow 3 in Figure 18(a) and Figure 19(f)). The robot can distinguish between traversable and nontraversable obstacles using proposed method and find the path to the goal position. The RRT planner searches the space previously given by the auxiliary A* path. It reduces the number of search trials that are necessary to find a path to the goal position. However, if the assessment about A* path is incorrect, the RRT planner still can find a path to the goal position. At the beginning of the experiment (Figure 19(a)), the RRT planner simultaneously explores the path on the right side of the Pioneer robot despite the fact that A* guides to the left side.

To demonstrate that the full body motion plans computed by the guided-RRT planner enable the legged robot to locomote over very difficult rough terrains, we prepared an experiment involving a rocky terrain mockup. The experimental environment is presented in Figure 20(a). Although this mockup has been successfully traversed by the Messor robot using a simpler motion planner (Belter & Skrzypczyński, 2011b), it is much more challenging to the smaller Messor II, which has shorter leg segments. Some of

the obstacles on the mockup are higher than the length of the last (tibia) segment. Therefore, whenever the configuration of the obstacles in the field of view of the robot permits such a solution, the guided-RRT planner avoids climbing the rocks altogether (Figure 20(b)) and takes a route round the mockup. The only way to force the robot to traverse the very challenging terrain is to set the goal position on the mockup or to add additional obstacles on the sides of the mockup, which are not traversable to the robot, and force the planner to yield a path over the rocky terrain (Figure 20(c)). However, the full body motion plan devised by the RRT algorithm successfully takes the Messor II robot over the mockup.

We used the new Messor II six-legged robot in the outdoor experiments. This robot is much more reliable mechanically and offers a more powerful onboard computer for the control algorithms. The remote operator, which uses a laptop computer, determines the goal position of the robot. The goal state is described by horizontal position and orientation of the robot. The full state of the robot is determined automatically by the motion planner.

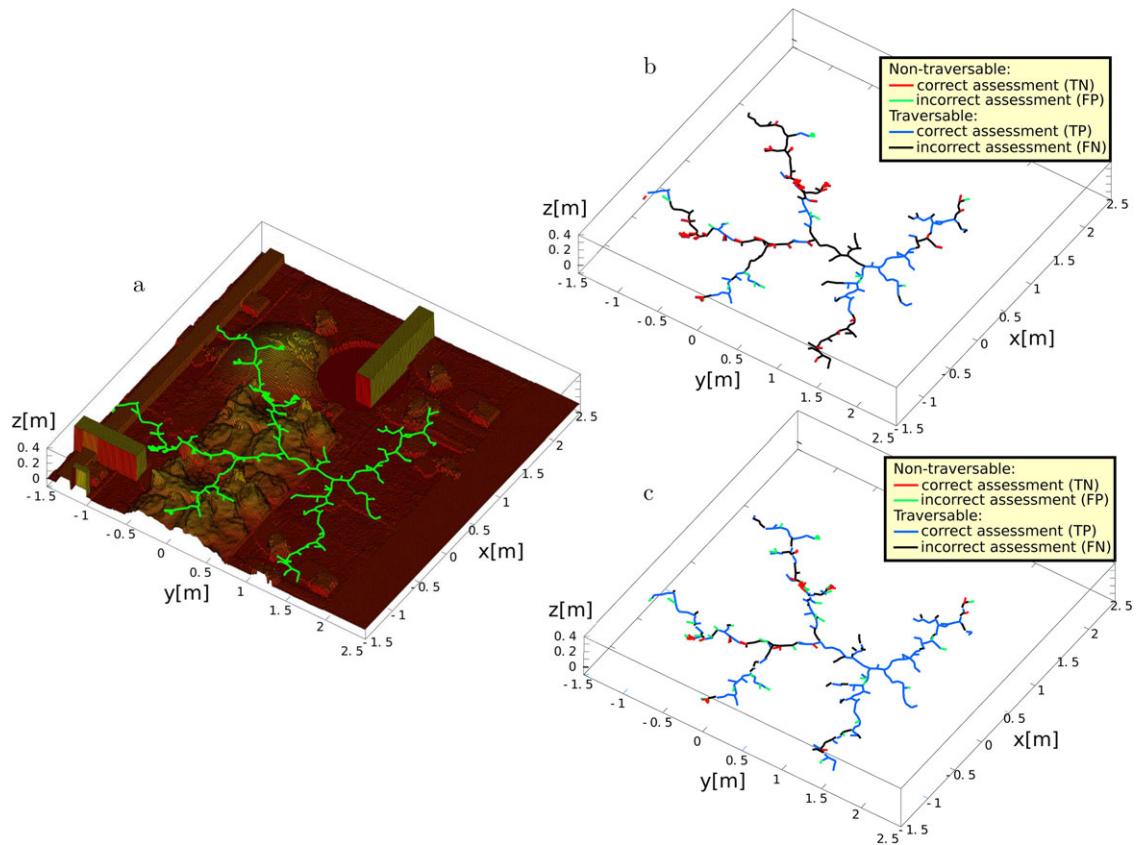


Figure 17. Traversability assessment before (b) and after prediction (c) – verification set.

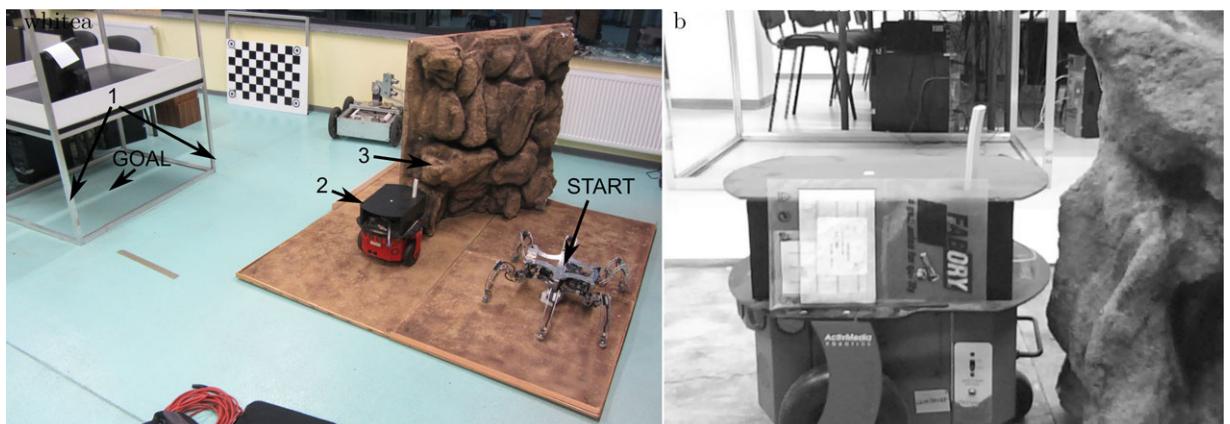


Figure 18. Indoor experiment verifying the integrated system (including the traversability assessment module) on the Messor robot: experimental setup (a) and camera view at the beginning of the experiment (b).

The Messor II robot is equipped with the Asus Xtion Pro sensor, which cannot operate in sunlight. Thus, we performed the experiments during cloudy weather. Despite that, the obtained range measurements are noisy and might contain some artifacts, or more commonly, “holes”

due to the spots of sunlight. Thus, we use two additional filters that are applied within a small window: a median filter that removes erroneous spikes of elevation, and a simple averaging filter that smoothes the resulting grid map.

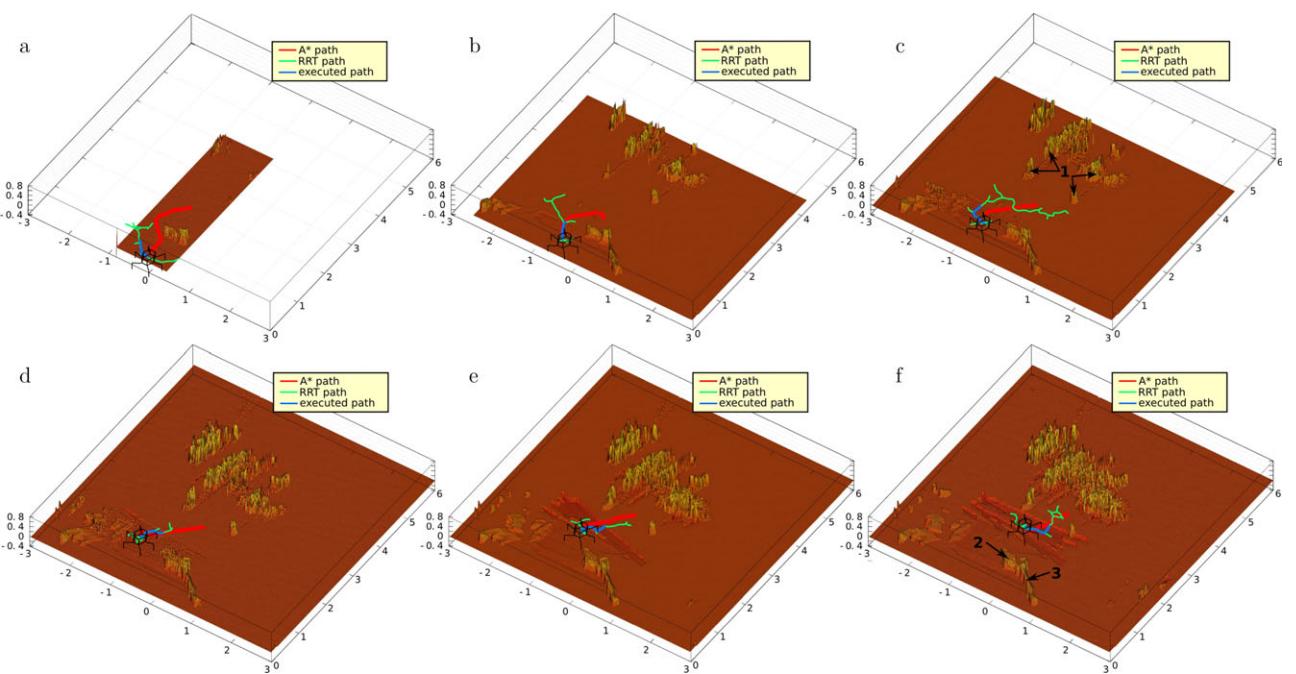


Figure 19. Indoor experiment verifying the integrated system on the Messor robot – subfigures (a)–(f) show the sequence of gradually updated elevation maps and motion plans including the coarse path, the RRT tree, and the final path.

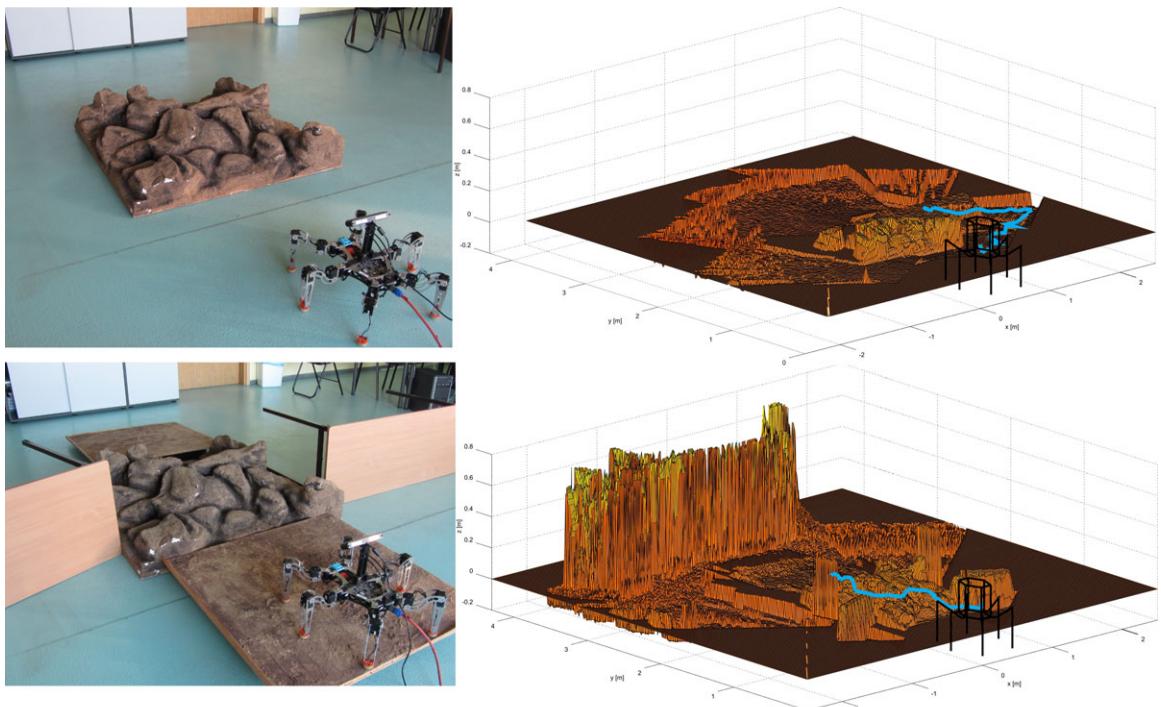


Figure 20. Results of an experiment on a rough-terrain mockup. The obtained paths depend on the configuration of the obstacles. With the guided-RRT planner running the robot avoids rough terrain mockup (b) and plans the path on flat terrain (a). With more complex environment (d) the robot is forced to climb the mockup (c).



Figure 21. Experimental site for the experiment performed on concrete and a pavement made of tiles.

The averaging filter is applied only when the difference between the highest and the lowest value of the elevation in the filter window is smaller than 0.25 m. It allows the preservation of walls, trees, and sharp edges of obstacles. Moreover, the minimal range of the Xtion sensor is 0.8 m and obstacles that are too close to the robot at the moment of measurement are invisible. Thus, we started each experiment on the relatively flat and unobstructed area. We also avoided having obstacles in front of the robot at the beginning of the experiment.

At first, we have verified the RRT-based motion planner in the environment shown in Figure 21. By defining the goal in front of the robot, not far from it, we “forced” the planner to plan a motion sequence through the barely traversable step near the corner of the wall (Figure 22). The experiment shows that the RRT-based precise motion planner can yield a sequence of motion that takes the robot through demanding obstacles, including steps and curbs higher



Figure 22. Motion planning without guidance by the coarse path from the A* algorithm in the environment shown in Figure 21: path found by the RRT-based planner (a), Messor II robot following this path (b), and a corresponding view of the elevation map and the planned motion sequence (c).

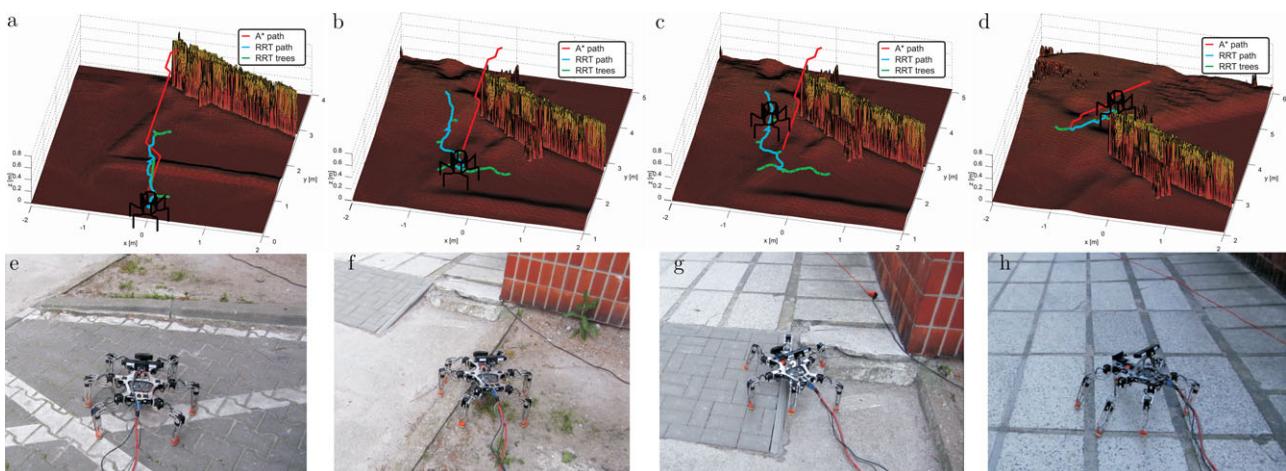


Figure 23. Results of an experiment on concrete/tiles, involving the Messor II robot with the fully integrated perception-based motion planning system. The subfigures (a) to (h) demonstrate the consecutive phases of path planning and motion execution.

than 50% of the length of the legs of Messor II (Figure 22(a) and (c)). However, such situations are risky from the point of view of the mechanical reliability of the robot (e.g., a servo might be broken). The proposed guided-RRT algorithm helps to avoid similar situations, in most cases a longer but safer detour to the RRT-based motion planner.

This is clearly demonstrated in Figure 23 – the robot operated in the same environment, but the start and goal poses were defined differently, giving the coarse path planner a chance to compute a reasonable path that avoided the highest part of the step. The goal pose of the robot was 5.5 m in front of the machine (in this experiment, the robot covered a path that was 18.33 times the length of the robot itself). To reach this goal, the robot had to make a detour from the straight path to avoid a wall. At the beginning of the experiment, the robot looked around to create an initial elevation map (Figure 23(a)). This procedure allowed the acquisition of a much better map, as a single snapshot from the Asus Xtion is not sufficient because of the narrow horizontal field of view of this sensor. The motion plans obtained from the guided-RRT algorithm as well as the gradually updated elevation maps corresponding to these plans are depicted in the subfigures from Figure 23(b) to Figure 23(h). One can see that the proposed motion planning method finds a path that goes around the most demanding obstacles – the robot prefers to go through the ramp. On the other hand, when required, the robot can find a way to deal with more demanding obstacles.

The length of the local RRT path executed by the robot is 2 m. After execution of the planned motion, the robot repeats the terrain perception procedure and motion planning. To obtain the goal position, the robot required three planning/replanning procedures. As expected, the robot avoids the wall to reach the goal. The robot follows the

general path provided by the A* planner. The second planning iteration results in a A* path (red path in Figure 23(b) and (c)) going through a concrete step (on the right of the robot in Figure 23(g)). In this case, the RRT-based motion planner found it difficult to follow the A* path and planned a shorter path through a ramp. Whenever the RRT-based planners finds some difficulties to determine the motion sequence to the reference pose, it starts to explore alternative solutions. In this case, the planner considers going to the right to reach the goal position (green branches of the RRT trees presented in Figure 23(b) and (c)). This solution is longer, so the RRT planner automatically chooses the path through the ramp.

Another outdoor experiment is performed in the environment that locally contains only small irregularities; however, unlike the previous scene, this one contains some larger obstacles, which the robot should avoid at the level of coarse path planning. The environment is presented in Figure 25. The environment contains a pebbly terrain and regularly shaped obstacles. The biggest challenge is related to the narrow passage, which is difficult to pass for a random-sampling-based planner, like the RRT. The robot has to make a bigger detour from the straight path to reach the goal. In this case, the path is executed at the distance 4.5 m. The planner requires two iterations to reach the goal position.

During the first iteration of the planning procedure (Figure 24(a)), the planner explores the environment to find the way through the corridor. However, the robot checks the possibilities of walking left and right round the obstacles (green branches of the tree represented by green lines in Figure 24(a) and (b)) it finds the path suggested initially by the A* planner. The planner explores also the area that is behind the wall in front of the robot. At the initial state, the robot does not have information about other walls of

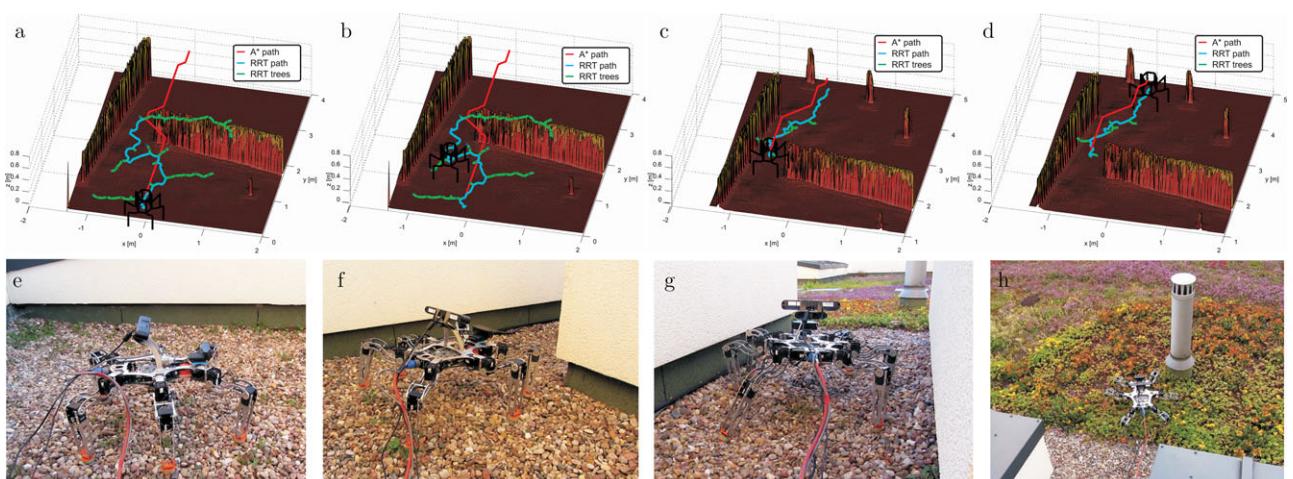


Figure 24. Experiment on the real robot using guided-RRT algorithm and proposed traversability assessment method.



Figure 25. Experimental set of the experiment on the pebbly terrain.

this object, as they are occluded due to the line-of-sight constraints. The algorithm can further correct the initial path whenever the robot acquires new information about the environment. In this case, it is not required, because the robot finds the path through the passage and goes straight to the goal that is next to the small chimney (Figure 24(d) and (h)). The ratio between the distance covered in this experiment by the Messor II robot and the length of its trunk is 16.7.

8. CONCLUSIONS

This article presents an integrated, perception-based motion planning system for statically stable walking robots (i.e., legged machines having six or more legs), which can be used in USAR missions. The main theoretical contribution of this work lies in

- a new, integrated motion planning approach for multi-DOF walking robots has been thoroughly presented, which combines an efficient implementation of the RRT-Connect planner paradigm for full-body motion planning in the presence of complex constraints imposed by the nature of a legged robot, and an adaptive version of the fast A* planner, which yields a coarse path, followed by the RRT-based planner;
- a new prediction-based terrain traversability assessment method, which allows the robot to predict how to assess terrain traversability given limited and imprecise map of the terrain by exploring this map with the RRT algorithm, and by using regression to estimate the traversability cost map; and
- an improved method for elevation mapping of rough terrain with a multisensor perception system of a walking robot, which introduces a new formulation of the elevation update rule that combines the simplicity of the Ye and Borenstein's method with explicit handling of

the elevation uncertainty by means of the probabilistic framework.

As the practical contribution, we show experimentally that by employing our motion planning concept we obtain motion sequences (plans) that are much better than the sequences we were able to compute using a simplistic adaptation of the RRT-Connect algorithm (Belter & Skrzypczyński, 2011a) or the nonadaptive variant of the guided-RRT method (Belter, 2011; Belter et al., 2011). This is due to the following:

- The A* algorithm allows in most cases to obtain a feasible coarse path to the goal; thus, the RRT-Connect algorithm that uses via points on this path as the input does not waste time and computing resources exploring areas unrelated to any feasible solution.
- The proposed guided-RRT formulation is insensitive to even large inaccuracy of the coarse path, because whenever the high-resolution local map is available, the motion planner locally checks all the constraints related to the terrain and the robot itself (kinematics, stability, etc.) and yields a new motion sequence, which can go round obstacles or traverse areas that were wrongly recognized as nontraversable.
- Although the higher-level, coarse path planner works with a low-resolution elevation map of the terrain, it uses a consistent traversability assessment procedure that takes into account that the local traversability is a function of both the local terrain geometry and the instantaneous kinematic configuration of the walking robot.

It was not possible to directly compare our motion planner to other approaches known from the literature, as each of these solutions is dedicated to a specific type/class of walking robot and involves different assumptions as to the type of motion (static gaits, dynamic gaits, bounding motion), and the amount of information that is available a priori to the robot, such as a perfectly known map and pose. However, the approach presented in this article is particularly minimalist as to the assumptions concerning the available prior data and any support the robot can receive from the operator. Although the Messor II robot we have used for the most extended outdoor experiments is not fully autonomous with regard to the onboard power and computing resources (an external laptop is used for planning), the system is completely autonomous with regard to the information processing: All the plans, at both the coarse and the full-body levels are devised only on the data acquired by the onboard sensors of the robot. The only input that is necessary from the human operator is to define the goal position. This makes our system rather unique, because as far as we know from the literature, all the perception-based motion planning systems demonstrated on walking robots were either limited to laboratory experiments (Kolter et al.,

2009; Satzinger et al., 2014; Shkolnik, 2010) or they used only coarse path planning and relied more on either clever mechanical design (Rusu et al., 2009) or on behavior-based control (Wooden et al., 2010). The motion planners for autonomous humanoids, like the one presented in Nishiwaki et al. (2012), have to deal with a significantly different problem: The biped is essentially only dynamically stable, which requires very fast and sophisticated walking control, but the discrete obstacles typically negotiated by humanoids in a man-made environment are much less complicated than the rough terrain. An implicit assumption commonly made in walking motion planning for humanoids is that the environment is simple enough to cause collisions only with the feet of the robot. We do not rely on such assumptions and take into account all possible collisions. Moreover, in a demanding terrain, we use a free gait; that is, we enable the robot to put its feet at the terrain positions that provide best support, not at those that were preplanned in some cyclic gait and constitute a fixed set (Belter, 2012). Although we use two algorithms A* and RRT that have been already applied in motion planning for legged robots, we use them in a different way than they have been used before. A variant of the RRT algorithm is used to plan the full-body motion because it is more efficient than A* in highly constrained search spaces (Perrin et al., 2012); however, the RRT planner is guided by a coarse path provided by the A*-based planner, avoiding unnecessary expansion of the tree. This can be considered as shifting the paradigm of A* search with nontrivial heuristics that improves its speed (Hornung et al., 2012) to a higher level, as the RRT algorithm does not require the guiding “heuristics” to be admissible: it can either make a detour if it finds the coarse path non-traversable or can make a shortcut, if the coarse path unnecessarily avoids some traversable area. It is also important that we let the RRT planner to “teach” the coarse A*-based planner how to make a proper traversability assessment. Owing to this adaptation, both planners collaborate efficiently without much human intervention and a necessity to gauge some arbitrary traversability parameters for each particular environment.

In this paper, we presented experimental evidence that the RRT planning paradigm may be applied efficiently in a motion planner reliable enough to be used in real field experiments. Although the scale of field experiments with the recently introduced LS3 robot is much bigger (Bajracharya et al., 2013), even taking into account the much larger size of this machine, the tests of LS3 were performed mostly in teleoperation mode. While they give good evidence as to the reliability of the terrain mapping system, the role of the planning component was much smaller than in our system, which autonomously planned and executed full-body motion sequences in nearly real-time. The somewhat limited scale of our experiments (relatively short paths) is mostly caused by the lack of full autonomy of our hardware platform – the Messor II robot is tethered to a power source

and an external laptop, which limits its mobility in natural terrain. Adding a high capacity, but lightweight battery should improve this situation, but it is a matter of technological progress, unrelated to the proposed navigation system.

Results of some of the experiments have shown also the limitations of our approach. The problems were mostly caused not by the planning components themselves, but by the limited reliability of perception with the onboard sensors. As it is not possible to use a 3D laser scanner on a small walking robot due to the tight mass and energy limits, we have to rely on compact sensors, which however, yield limited quality range data containing many artifacts. Alternatively, we have demonstrated, that it is possible to use even outdoors a popular structured light RGB-D sensor, like the Asus Xtion. This solution may be attractive to other researchers, but in real USAR missions, sensors of higher reliability are necessary. Another factor that limits our system is the dependency on precise self-localization. We have shown elsewhere (Belter & Skrzypczyński, 2013) that it is possible to employ PTAM on a walking robot, obtaining pose estimates that are precise enough for elevation mapping, foothold selection, and so forth, but in some environments, the passive-vision-based solution fails easily. This was the case during the experiments on the terrain covered by pebbles, which caused PTAM to diverge quickly due the multitude of perceptually indistinguishable point features. Therefore, we are researching in parallel other approaches to “lightweight” self-localization on a walking robot (Nowicki & Skrzypczyński, 2013).

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The videos are available as Supporting Information in the online version of this article.

Extension	Media type	Description
1	Video	Simulation experiment
2	Video	Indoor experiment
3	Video	Outdoor experiment
4	Video	Experiment on the rough terrain mockup

ACKNOWLEDGMENTS

This work was financed by the National Science Center under decision DEC-2011/01/N/ST7/02080.

REFERENCES

- Arain, A., Havoutis, I., Semini, C., Buchli, J., & D. C. (2013). A comparison of search-based planners for a legged robot. In IEEE International Workshop on Robot Motion and Control, Wasowo, Poland. July 3–5.

- Bai, S., & Low, K. (2001). Terrain evaluation and its application to path planning for walking machines. *Advanced Robotics*, 15(9), 729–748.
- Bajracharya, M., Ma, J., Malchano, M., Perkins, A., Rizzi, A., & Matthies, L. (2013). High fidelity day/night stereo mapping with vegetation and negative obstacle detection for vision-in-the-loop walking. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3663–3670. Tokyo, Japan. November 3–7.
- Belter, D. (2011). Perception-based motion planning for a walking robot in rugged terrain. In Lecture Notes in Control and Information Sciences: Robot Motion and Control (K. Kozłowski, (Ed.)), Berlin: Springer. pp. 127–136.
- Belter, D. (2012). Gait modification strategy for a six-legged robot walking on rough terrain. In A. Azad et al., editors, *Adaptive Mobile Robotics*, pages 367–374, Singapore: World Scientific.
- Belter, D. (2013). Optimization-based approach for motion planning of a robot walking on rough terrain. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 7(4), 34–41.
- Belter, D., Łabęcki, P., & Skrzypczyński, P. (2010). Map-based adaptive foothold planning for unstructured terrain walking. In Proceedings of the IEEE International Conference on Robotics and Automatization, pp. 5256–5261, Anchorage. May 3–8.
- Belter, D., Łabęcki, P., & Skrzypczyński, P. (2011). On-board perception and motion planning for legged locomotion over rough terrain. In 5th European Conference on Mobile Robots, pages 195–200, Örebro, Sweden. September 7–9.
- Belter, D., Łabęcki, P., & Skrzypczyński, P. (2012). Estimating terrain elevation maps from sparse and uncertain multi-sensor data. In IEEE 2012 International Conference on Robotics and Biomimetics, pages 714–722, Guangzhou, China. December 11–14.
- Belter, D., Łabęcki, P., & Skrzypczyński, P. (2013). An exploration-based approach to terrain traversability assessment for a walking robot. In 11th IEEE International Symposium on Safety Security and Rescue Robotics, pages 1–6, Linköping, Sweden. October 21–26.
- Belter, D., & Skrzypczyński, P. (2010). A biologically inspired approach to feasible gait learning for a hexapod robot. *International Journal of Applied Mathematics and Computer Science*, 20(1), 69–84.
- Belter, D., & Skrzypczyński, P. (2011a). Integrated motion planning for a hexapod robot walking on rough terrain. In 18th IFAC World Congress, pages 6918–6923, Milan, Italy. August 28 – September 2.
- Belter, D., & Skrzypczyński, P. (2011b). Rough terrain mapping and classification for foothold selection in a walking robot. *Journal of Field Robotics*, 28(4), 497–528.
- Belter, D., & Skrzypczyński, P. (2012). Posture optimization strategy for a statically stable robot traversing rough terrain. In IEEE/RSJ 2012 International Conference on Intelligent Robots and Systems, pages 2204–2209, Vilamoura, Portugal. October 7–12.
- Belter, D., & Skrzypczyński, P. (2013). Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping. *Industrial Robot: An International Journal*, 40(3), 229–237.
- Belter, D., & Walas, K. (2014). A compact walking robot – flexible research and development platform. In R. Szewczyk et al., Eds., *Recent Advances in Automation, Robotics and Measuring Techniques*, volume 267 of *AISC*, pages 343–352, Springer, Zürich.
- Bonnafous, D., Lacroix, S., & Siméon, T. (2001). Motion generation for a rover on rough terrains. In IEEE/RSJ Int. Conf. on Intell. Robots and Systems, pages 784–789, Maui, Hawaii. October 29 – November 3.
- Chestnutt, J. (2013). Navigation and gait planning. In Harada, K., Yoshida, E., and Yokoi, K., editors, *Motion Planning for Humanoid Robots*, London: Springer.
- Chestnutt, J., Takaoka, Y., Suga, K., Nishiaki, K., Kuffner, J., & Kagami, S. (2009). Biped navigation in rough environments using on-board sensing. In IEEE/RSJ 2009 International Conference on Intelligent Robots and Systems, St. Louis, USA, , October 11–15 pages 3543–3548.
- Cupec, R., Schmidt, G., & Lorch, O. (2005). Vision-guided walking in a structured indoor scenario. *Automatika*, 46, 49–57.
- Dryanovski, I., Morris, W., & Xiao, J. (2010). Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles. In Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Systems, pages 1553–1559, Taipei, Taiwan ROC. October 18–22.
- Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1), 32–40.
- Hansen, E. B. (2004). Nonparametric conditional density estimation. *Neural Computation*, 21(2), 533–559.
- Hauser, K., Bretl, T., Latombe, J.-C., Harada, K., & Wilcox, B. (2008). Motion planning for legged robots on varied terrain. *International Journal of Robotics Research*, 27(11–12), 1325–1349.
- Hauser, K., Bretl, T., Latombe, J.-C., & Wilcox, D. (2006). Motion planning for a six-legged lunar robot. In 7th Int. Workshop on the Algorithmic Foundations of Robotics, pages 301–316, New York, USA. July 16–18.
- Hoepflinger, M., Hutter, M., Gehring, C., Bloesch, M., & Siegwart, R. (2014). Haptic foothold suitability identification and prediction for legged robots. In et al., M. O. T., editor, *Mobile Service Robotics*, pages 425–432. Singapore: World Scientific.
- Hornung, A., Dornbush, A., Likhachev, M., & Bennewitz, M. (2012). Anytime search-based footstep planning with sub-optimality bounds. In IEEE-RAS International Conference on Humanoid Robots, pages 674–679, Osaka, Japan. 29 Nov – 01 Dec.
- Howard, A., & Seraji, H. (2001). Vision-based terrain characterization and traversability assessment. *Journal of Robotic Systems*, 18(10), 577–587.
- Jaillet, L., Cortes, J., & Simeon, T. (2008). Transition-based rrt for path planning in continuous cost spaces. In IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, pages 2145–2150, Nice, France. September 22–26.

- Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., & Schaal, S. (2010). Fast, robust quadruped locomotion over challenging terrain. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 1050–4729, Anchorage, May 3–8.
- Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics & Automation*, 12(4), 566–580.
- Kelley, C. (1999). Iterative methods for optimization. SIAM, Philadelphia.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings IEEE International Conference on Neural Networks, pages 1942–1948, Piscataway, Australia.
- Khoskelham, K., & Elberink, S. O. (2012). Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12, 1437–1454.
- Kirby, R., Simmons, R., & Forlizzi, J. (2009). Variable sized grid cells for rapid replanning in dynamic environment. In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4913–4918, St Louis, October 11–15.
- Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In Proceedings International Symposium on Mixed and Augmented Reality, pages 225–234, Nara, Japan, May 12–17.
- Kolter, J., Kim, Y., & Ng, A. (2009). Stereo vision and terrain modeling for quadruped robots. In Proceedings IEEE International Conference on Robotics and Automation, pages 3894–3901, Kobe, Japan.
- Kolter, J., Rodgers, M., & Ng, A. (2008). A control architecture for quadruped locomotion over rough terrain. In Proceedings IEEE International Conference on Robotics and Automation, pages 811–818, Pasadena, USA, May 19–23.
- Konolige, K. (1997). Small vision systems: Hardware and implementation. In Proceedings Eighth International Symposium on Robotics Research, pages 111–116, Hayama, Japan, October 4–7.
- Krotkov, E., & Hoffman, R. (1994). Terrain mapping for a walking planetary rover. *IEEE Transactions on Robotics and Automation*, 10(6), 728–739.
- Krotkov, E., & Simmons, R. (1996). Perception, planning, and control for autonomous walking with the Ambler planetary rover. *Int. Journal of Robotics Research*, 15(2), 155–180.
- Kubota, T., Kuroda, Y., Kunii, Y., & Yoshimitsu, T. (2001). Path planning for newly developed microrover. In Proceedings IEEE/RSJ International Conference on Robotics and Automation, pages 3710–3715, Seoul, Korea, May 21–26.
- Kuffner, J., Kagami, S., Nishiwaki, K., Inaba, M., & Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1), 105–118.
- Kumar, P. D., Kalyanmoy, D., & Amitabha, G. (2002). Optimal path and gait generations simultaneously of a six-legged robot using a GA-fuzzy approach. *Robotics and Autonomous Systems*, 41(1), 1–20.
- LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Iowa State University, USA.
- LaValle, S. M., & Kuffner, J. J. (2001). Rapidly-exploring random trees: Progress and prospects. In Donald, B. R., editor, *Algorithmic and Computational Robotics: New Directions*, pages 293–308, Wellesley, A. K. Peters/CRC Press.
- Lee, J., Pippin, C., & Balch, T. (2008). Cost based planning with RRT in outdoor environments. In IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, pages 684–689, Nice, France, September 22–26.
- Lindemann, S. R., & LaValle, S. M. (2004). Steps toward de-randomizing RRTs. In Proceedings of the 4th International Workshop on Robot Motion and Control, pages 271–277, Puszczynkowo June 17–20.
- Kuffner Jr., J. J., & LaValle, S. M. (2000). RRT-Connect: An efficient approach to single-query path planning. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 995–1001, San Francisco, April 24–28.
- Łabęcki, P. (2011). Improved data processing for an embedded stereo vision system of an inspection robot. In T. Březina, R. J., editor, *Mechatronics: Recent Technological and Scientific Advances*, pages 749–757. Springer.
- Łabęcki, P., & Belter, D. (2013). Sensory system calibration method for a walking robot. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 7(2), 39–45.
- Łabęcki, P., Rosiński, D., & Skrzypczyński, P. (2011). Terrain map building for a walking robot equipped with an active 2d range sensor. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 5(3), 67–78.
- Łabęcki, P., & Skrzypczyński, P. (2013). Spatial uncertainty assessment in visual terrain perception for a mobile robot. In J. Korbicz, M. K., editor, *Intelligent Systems in Technical and Medical Diagnostics*, AISC Vol. 230, pages 357–368, Berlin, Springer.
- McGhee, R. B., & Iswandhi, G. (1979). Adaptive locomotion for a multilegged robot over rough terrain. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-9(4), 176–182.
- Murphy, R. (2014). *Disaster Robotics*. Cambridge, MA: MIT Press.
- Nishiwaki, K., Chestnutt, J., and Kagami, S. (2012). Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor. *International Journal of Robotics Research*, 31(11), 1251–1262.
- Norouzi, M., Miro, J. V., & Dissanayake, G. (2012). Planning high-visibility stable paths for reconfigurable robots on uneven terrain. In IEEE/RSJ 2012 International Conference on Intelligent Robots and Systems, pages 2844–2849, Vilamoura, Portugal, October 7–12.
- Nowicki, M., & Skrzypczyński, P. (2013). Combining photometric and depth data for lightweight and robust visual odometry. In Proceedings of the European Conference on Mobile Robots, pages 125–130, Barcelona, Spain, September 25–27.
- Okubo, Y., Ye, C., & Borenstein, J. (2009). Characterization of the Hokuyo URG-04lx laser rangefinder for mobile robot

- obstacle negotiation. In Unmanned Systems Technology XI, Proceedings SPIE 7332.
- Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4), 1373–1385.
- Papadakis, P., & Pirri, F. (2012). 3d mobility learning and regression of articulated, tracked robotic vehicles by physics-based optimization. In *Workshop on Virtual Reality Interactions and Physical Simulations*, Eurographics, pages 147–156. Darmstadt, Germany, December 6–7.
- Park, J.-H., Shin, Y.-D., Bae, J.-H., & Baeg, M.-H. (2012). Spatial uncertainty model for visual features using a Kinect sensor. *Sensors*, 12, 8640–8662.
- Perrin, N., Stasse, O., Baudouin, L., Lamiriaux, F., & Yoshida, E. (2012). Fast humanoid robot collision-free footstep planning using swept volume approximations. *IEEE Transactions on Robotics*, 28(2), 427–439.
- Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N., & Burgard, W. (2009). A Bayesian regression approach to terrain mapping and an application to legged robot locomotion. *Journal of Field Robotics*, 26(10), 789–811.
- Roennau, A., Heppner, G., Nowicki, M., & Dillmann, R. (2014). LAURON V: A versatile six-legged walking robot with advanced maneuverability. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 82–87. Besançon, France, July 8–11.
- Roennau, A., Kerscher, T., Ziegenmeyer, M., Marius, J., Zolner, J. M., & Dillman, R. (2009). Adaptation of a six-legged walking robot to its local environment. In Kozłowski, K., editor, *Robot Motion and Control 2009*, LNCIS Vol. 396, pages 155–164. Springer. London.
- Rusu, R. B., Sundaresan, A., Morisset, B., Hauser, K., Agrawal, M., Latombe, J.-C., & Beetz, M. (2009). Leaving flatland: Efficient real-time 3D perception and motion planning. *Journal of Field Robotics*, 26(10), 841–862.
- Sanchez, L. A., Zapata, R., & Osorio, L. M. (2008). Sampling-based motion planning: A survey. *Computacion y Sistemas*, 12(1), 5–24.
- Sanctis, L., Garrido, S., Moreno, L., & Blanco, D. (2009). Outdoor motion planning using fast marching. In O. Tosun et al., editors, *Mobile Robotics: Solutions and Challenges*, pages 1071–1080, Singapore: World Scientific.
- Satzinger, B., Reid, J., Bajracharya, M., Hebert, P., & Byl, K. (2014). More solutions means more problems: Resolving kinematic redundancy in robot locomotion on complex terrain. In *IEEE/RSJ 2014 International Conference on Intelligent Robots and Systems*, Boston, USA September 14–18, pp. 4861–4867, Chicago, USA.
- Shirkhodaie, A., Amrani, R., & Tunstel, E. (2005). Soft computing for visual traversability assessment by terrain perception and planetary robotic systems. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1848–1855, Hawaii, USA.
- Shkolnik, A. (2010). Sample-based motion planning in high-dimensional and differentially-constrained systems. Technical report, PhD Dissertation, MIT, Cambridge, MA.
- Shneier, M., Chang, T., Hong, T., Shackleford, W., Bostelman, R., & Albus, J. (2008). Learning traversability models for autonomous mobile vehicles, autonomous robots. *Autonomous Robots*, 24(1), 69–86.
- Smith, R. (2014). Open dynamics engine, <http://www.ode.org>.
- Stelzer, A., Hirschmüller, H., & Görner, M. (2012). Stereovision-based navigation of a six-legged walking robot in unknown rough terrain. *International Journal of Robotics Research*, 31(4), 382–402.
- Vernaza, P., Likhachev, M., Bhattacharya, S., Chitta, S., Kushleyev, A., & Lee, D. (2009). Search-based planning for a legged robot over rough terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2380–2387, Kobe, Japan, May 12–17.
- Vonasek, V., Faigl, J., Krajnik, T., & Preucil, L. (2009). RRT-path – a guided rapidly exploring random tree. In K. Kozłowski, editor, *Lecture Notes in Control and Information Sciences: Robot Motion and Control*, pages 307–316, Berlin: Springer.
- Walas, K., Belter, D., & Kasiński, A. (2008). Control and environment sensing system for a six-legged robot. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 2(3), 26–31.
- Walas, K., & Nowicki, M. (2014). Terrain classification using Laser Range Finder. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5003–5009. September 14–18, pp. 4861–4867, Chicago, USA.
- Wooden, D., Malchano, M., Blankepoor, K., Howard, A., Rizzi, A., & Raibert, M. (2010). Autonomous navigation for BigDog. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4736–4741, Anchorage, USA, May 3–8.
- Yamaguchi, A., Takamatsu, J., & Ogasawara, T. (2013a). Dcob: Action space for reinforcement learning of high dof robots. *Autonomous Robots*, 34, 327–346.
- Yamaguchi, A., Takamatsu, J., & Ogasawara, T. (2013b). Learning strategy fusion for acquiring crawling behavior in multiple environments. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 605–612, Shenzhen, China, December 12–14.
- Ye, C., & Borenstein, J. (2004). A novel filter for terrain mapping with laser rangefinders. *IEEE Transactions on Robotics and Automation*, 20(5), 913–921.