2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Neural Network Based Heterogeneous Sensor Fusion for Robot Motion Planning

Bijo Sebastian, Hailin Ren, and Pinhas Ben-Tzvi

Department of Mechanical Engineering
Virginia Tech
Blacksburg, Virginia 24060, USA

bentzvi@vt.edu

Abstract – **This paper presents a neural network based heterogeneous sensor fusion approach towards real-time traversability estimation of mobile robots using sensor data. Even though significant advances have been made for autonomous navigation in structured terrain conditions, obtaining reliable traversability estimates for tracked vehicle navigation in challenging terrain conditions is still an open research problem. In this regard, we propose a neural network architecture capable of fusing depth images along with roll and pitch measurements on board the robot to perform traversability estimation. The proposed architecture is trained in a variety of simulated structured and unstructured environments. As such, the proposed architecture is capable of extracting the relevant features from the sensor measurements in a data driven manner as compared to existing heuristic based approaches that fail to generalize for different environmental conditions. The reliability of the traversability estimates provided by the trained architecture was validated in indoor and outdoor conditions using real sensor data. In addition, the feasibility of using the traversability estimates in incremental path planning was also demonstrated through simulation. For both applications the proposed approach provided compelling results. Inferences based on the results of the experiments along with directions for future research are also outlined.**

Index Terms – Traversability, mobile robot, planning, neural network

I. Introduction

Autonomous navigation is a ubiquitous task for mobile robots. Great advances have been made in this domain, as demonstrated by the recent developments in self-driving vehicles, warehouse automation, and even smart vacuum systems that are now a common part of the households. Despite these advances, autonomous navigation in its true sense is still an open research problem for many challenging conditions such as tracked locomotion in unstructured terrain, which is the focus of this work.

For car-like or Hilare [1] type robots moving in structured terrain, the presence or absence of obstacles can be used to determine traversability in a trivial manner. In comparison to structured environments, tracked vehicle locomotion in unstructured terrain offers additional challenges. Several factors such as the characteristics of the terrain in terms of slip, slope, and soil properties, characteristics of the robot in terms of weight and moment of inertia, actuator limitations, and the nature of the track profile all play major role in determining traversability. In addition, tracked robotic systems are capable of navigation over most obstacles owing to their inherent mechanical advantages. As such, trivial estimation of traversability based on the presence or absence of obstacles will be an overly conservative approach for tracked systems.

All of the above factors make real-time traversability estimation using onboard sensor data a non-trivial problem for tracked systems. At the same time, estimating traversability is particularly important when it comes to reliable motion planning. For applications such as planetary exploration, it is mission critical to enable the system to obtain, understand, and utilize terrain information in real-time [2].

A majority of the existing traversability estimation methods [2]–[4] involve heuristic based techniques that rely on a variety of factors such as the maximum height of features in the terrain, slope, roughness, or even a combination of these as obtained from onboard sensors such as LIDAR, camera, Inertial Measurement Units (IMU), etc. These techniques are often specifically designed for the application at hand and do not generalize well for different environmental conditions. Others have used a full dynamic model of the robot along with the terrain map to determine the same [5], [6]. Even though they provide reliable results, full dynamic simulation of robot motion could be computationally intensive for the limited resources available onboard the robots. A detailed survey of existing geometry based and vision based terrain traversability methods is presented by Papadakis in [7].

Traditional techniques as mentioned above require the researcher to explicitly define the relationship between the detected features from the sensor data and the traversability of the terrain. This is difficult due to the fact that the dynamic interactions between the robot and the terrain are too complex to accurately model. To address this issue, recently there has been more interest in learning based traversability estimation techniques that are capable of deriving the relationship from the given data. Based on the diversity of the collected data, these techniques are capable of generalizing to environments with varied features and characteristics. In addition, even though the training process itself is computationally intensive, the trained machine learning (ML) architecture could be deployed with minimum hardware requirements, thereby improving the real-time applicability of these techniques.

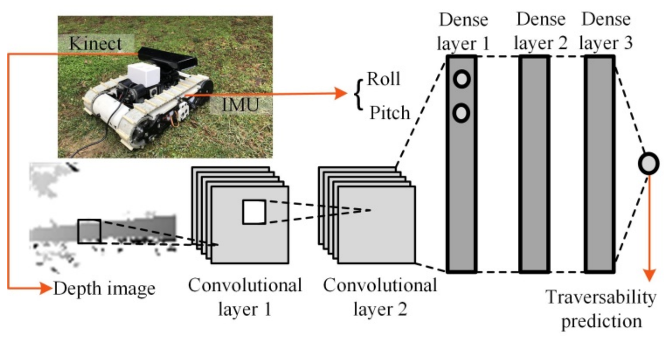

Fig.1 Overall layout of the proposed neural network architecture.

In this regard, Murphy et al. [8] presented a Gaussian process (GP) technique to predict the traversability of unknown locations based on traversability estimates garnered from onboard sensing along with a-priori available overhead color images of the regions. Chavez-Garcia et al. [9] presented traversability estimation as a heightmap classification problem that could be solved using a trained convolutional neural network (CNN) based on terrain patches represented as images. Unsupervised classification of outdoor 3D LIDAR data for future terrain traversability estimations was presented by Maligo and Lacroixin [10]. A semi supervised learning approach, where the robot learns its traversability capabilities from a human operator was presented by Suger et al. in [11]. An approach towards correlating exteroceptive (terrain observations or images) and proprioceptive (acceleration signals) information to assess terrain manoeuvrability for mobile robots was presented by Bekhti et al. in [12], whereas proprioceptive sensing information including wheel slip, vehicle orientation, vibration, and power consumption were used to map instantaneous traversability in [13]. An approach towards combining separate traversability maps generated from colour images and point clouds using Bayes' rule was presented by Sock et al. in [14]. A detailed survey of recent learning based techniques towards estimating terrain traversability is given by Wong et al. in [2].

II. Proposed Technique

This paper presents a novel learning based approach towards traversability estimation by using a neural network (NN) capable of combining heterogeneous sensor information to analyze terrain traversability in real-time. The proposed architecture consists of two convolutional layers and three dense layers as shown in Fig. 1. The convolutional layers operate on the depth image obtained from the Kinect sensor, while the dense layers operate on the output of the convolution layers stacked with the roll and pitch angles of the robot to generate traversability estimation. In comparison to existing state-of-the-art traversability estimation techniques, the proposed work offers the following novelties and advantages:

<u>(1) The ability to combine heterogeneous sensor data:</u> The proposed approach uses depth data of the robot's surroundings along with roll and pitch of the platform as input to the NN architecture. Even though there are previous approaches towards combining separate traversability maps from different sensing modalities [14] and also for correlating different sensor inputs [12], to the authors knowledge this work is the first attempt towards training a single neural network to use heterogeneous sensor data for estimating terrain traversability.

<u>(2) Prior knowledge of the environment is not required:</u> Existing learning based terrain traversability approaches in the literature require prior knowledge of the environment in some form, such as 3D maps [9], or overhead images [8]. There are many real-life applications of field robotic systems where prior information is not available. As mentioned by Shan et al. [3], assuming the prior availability of a 3D map can severely limit the reliability of these techniques when it comes to field robotic applications. The proposed approach on the other hand only relies on real-time sensor data using onboard sensors.

<u>(3) Use of simulations for generating training data:</u> Unlike existing approaches [10],[11] that use experimental data for training the traversability estimation techniques, the proposed approach uses training data generated from a high fidelity simulation. A robotic simulator capable of accurately modelling the dynamic motion of a mobile robot over challenging terrain conditions along with the sensor outputs was used. This allowed generating a diverse training dataset without any damage to the real robot, while ensuring that the trained policies are applicable to real-life conditions.

It should be noted that, even though combining sensory inputs inside a neural network and using simulations to generate training data has been explored previously in the machine learning community; this is the first time the above approaches are being used for traversability estimation. The remainder of the paper is organized as follows; Section III describes the use of robotic simulators to obtain accurate and extensive training data for the NN architecture. Section IV describes the proposed NN architecture and the methodology to combine roll and pitch data along with depth images obtained from a Kinect sensor. Section V describes the experimental validation of the proposed technique in performing traversability estimation in real-life conditions. Section VI validates the applicability of terrain traversability estimations in incremental path planning over unstructured terrain conditions outside of the training environment. Section VII concludes the paper with directions for future research.

III. Data Collection in Simulated Environment

One of the major difficulties with the use of ML based methods in robotics is the need for extensive and accurate datasets. In order to accurately judge whether or not a particular maneuver can be performed, the robot must actually perform the maneuver. Since the dataset will need samples of both feasible and non-feasible cases, the robot will have to perform maneuvers that could cause potential damage. In order to create an extensive and accurate dataset without damaging the robot, a robotic simulator, V-REP (Virtual Robotics Experimentation Platform) [15] was used to perform the desired maneuvers and record the results along with the simulated sensor data.

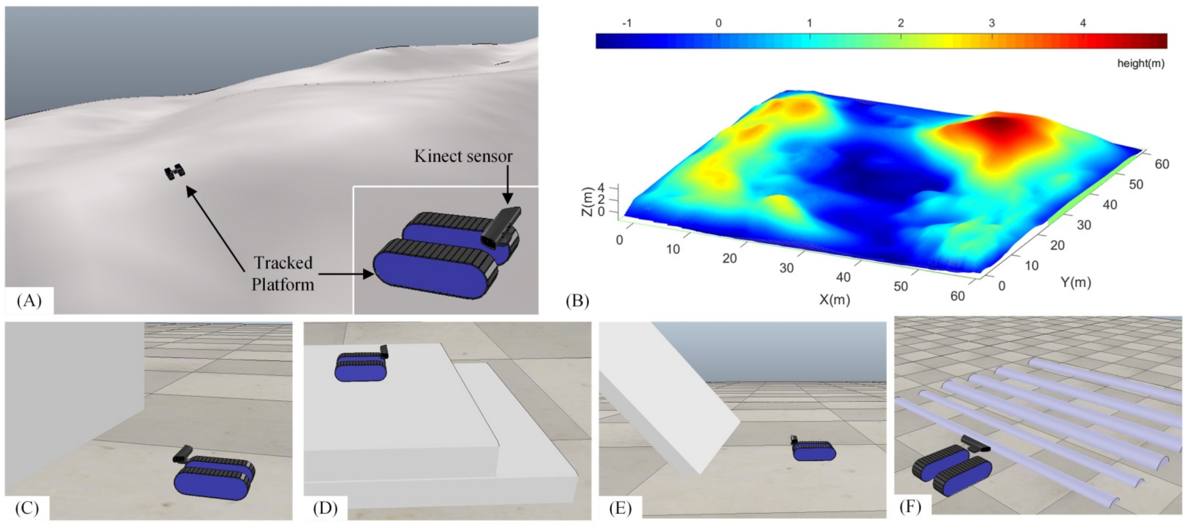

Fig. 2 Data collection through simulation: (A) Tracked platform with Kinect on simulated outdoor terrain, (B) Topography map of the terrain used. Simulated indoor scenarios (C) Facing a wall, (D) Facing edge of platform, (E) Facing inclined ramp, (F) Facing bumps of varying size.

V-REP is a robotic simulator by Coppelia Robotics. This cross platform software allows simulating robotic platforms including a variety of sensors and actuators, along with realistic physics engines that support dynamic simulations of robot motion. A tracked robotic platform with a Kinect depth sensor attached to its front was made to travel along a simulated terrain condition consisting of hills, valleys, and structured terrain consisting of walls, ramps, and bumps as shown in Fig. 2. The virtual terrain of size 60x60 m for the simulation was created in Blender using the Ant plugin [16]. Due to the varying terrain conditions, the frictional resistance experienced by the tracked robotic vehicle is not the same on both sides. As such, providing the same velocity command to both tracks does not guarantee that the robot will drive straight, even in the simulator. This makes it necessary to have a closed-loop controller running during the data collection simulations. Moreover, the same low-level controller is used on the actual robot for the experimental validation. As such, simulating the controller during data collection makes the data more representative of the actual motion of the robot in real-life conditions.

The motion of the robot in the simulation was performed using a low-level controller that assumed a unicycle robot model as given by

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where (x, y) is the 2D position and θ represents the orientation of the robot fixed frame with respect to the global inertial frame, ω is the angular velocity of the robot, and V is the linear velocity. Assuming the goal location is at the coordinates (x_g, y_g), the desired orientation of the robot is given by:

$$\theta^* = \tan^{-1}\left(\frac{y_g - y}{x_g - x}\right) \tag{2}$$

The error in orientation is given by $e = \theta^* \ominus \theta$ where $\ominus$ denotes difference taking into account the wraparound of angles. The PD (Proportional-Derivative) controller for the angular velocity of the robot is given by:

$$\omega = k_p e + k_d \dot{e}. \tag{3}$$

where k_p and k_d are the proportional and derivative gains. The linear velocity of the robot is scaled based on ω so that the robot slows down before making sharp turns. This allows for a smoother navigation. The maximum linear and angular velocity commands from the controller were limited to 0.03 m/s and 1.0 rad/s, respectively. Once the desired linear and angular velocities have been calculated using the low level controller, they can be transformed into the desired left and right track velocities using the transformation equations between differential drive and unicycle robot models before they are applied to the tracked robot.

It should be noted that even though the low-level controller presented here is based on robot kinematics, V-REP performs a full dynamic simulation of the robot motion at each instant. Before conducting the data collection simulation, the weight, size and actuation limits of the simulated robot were adjusted to be the same as that of the hardware used for experimental validation, STORM [17]. The simulated robot was 0.41 m in length, 0.3 m in width and 0.12 m in height with a total weight of 9 kg, similar to the actual robot. The torques applied on the tracks were limited to a maximum of 10 Nm with a PID (Proportional-Integral-Derivative) speed controller simulating the performance of the motor driver on the actual robot. This further reduced the possible mismatches between the behavior of the simulated and real robot.

For the purpose of data collection, the low-level controller was implemented in MATLAB and was made to communicate with V-REP. Based on the simulated robot pose (x, y, θ) as obtained from V-REP, the desired control commands were sent back from MATLAB during the simulation.

In order to create the training instances, the simulated robot was placed at random poses on the simulated terrain

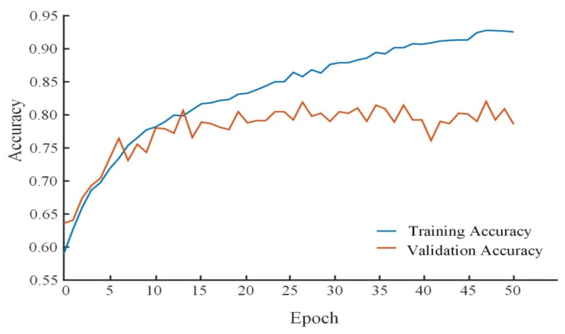

Fig. 3 Variation of the training and validation accuracy with epochs.

and checked for static stability. Once the robot was stable, a depth image of the area directly in front of the robot was obtained from the simulated Kinect sensor, along with the roll and pitch of the robot. This data was sent to MATLAB and stored. The robot was then required to move towards a goal location directly in front of the robot at a fixed distance of 1m using the closed-loop controller. If the robot was able to reach within 0.1m of the goal location within 75 seconds of simulation time, the motion was recorded as successful, and if not, as a failure.

Ten thousand instances of robot motion were simulated to create an extensive and accurate dataset. This dataset was used as training and test data for the proposed NN architecture, as shown in Fig. 1. Each training instance consists of a depth image (48x64 array) of the region in front of the robot, roll and pitch (normalized values) of the robot before the motion was started, and the result of whether the motion was successfully completed.

IV. Neural Network (NN) Architecture

This section explains the proposed NN architecture which incorporates the depth image and the IMU data to predict traversability of the terrain. The proposed architecture consists of two convolutional layers and three dense fully connected layers.

The convolutional layers process the depth images to extract features that correspond to the traversability of the terrain. Each convolutional layer consists of 32 filters with 2x2 kernel size followed by a max pooling layer with the same kernel size. For the proposed architecture, the convolution kernel is regressed inherently as part of the training process. IMU data is fed into the network through the first fully connected layer. This is done by concatenating the IMU data with the flattened output of the second convolutional layer in order to improve the accuracy of the final prediction. Each fully connected layer consists of 256 neurons except for the last one, which consists of only one neuron to perform the traversability prediction. A sigmoid activation function is used in the prediction layer while all the other layers use leaky ReLU activation function [18], as given by

$$f(x) = \begin{cases} x & x > 0 \\ 0.02x & otherwise \end{cases} \quad (4)$$

The neural network is built and trained in Keras with TensorFlow [19] as the backend using Adam optimizer [20]. A block diagram representation of the proposed neural network architecture is provided in Fig. 1.

In order to prevent scaling of the weights based on the varying units of the incoming heterogeneous data, normalization was performed on the raw input data before being fed into the NN. The raw depth image was clipped to measure only between 0m to 4m and then normalized between [0,1]. The IMU data was clipped between [-40, 40] degrees and normalized within [0, 1].

To train the NN, the dataset was split randomly: 9,000 samples were used for training and 1,000 samples for validation. Both training and validation datasets contained approximately half traversable and half non-traversable samples. The batch size used for training was 128 and the learning rate was set to 0.001. Adam optimizer [20] in Keras was used for the training. As shown in Fig. 3, the proposed architecture gave an average validation accuracy of 80% after ten epochs, with the lowest loss on Epoch 17. To avoid overfitting, the best weights of the NN were saved by monitoring the minimum loss of validation during the training process. The trained architecture provided 91.6% accuracy on detecting true positives (traversable) and 72.0% accuracy on detecting true negatives (non-traversable).

V. Experimental Validation

One of the inherent issues with training ML architectures inside simulations is that the real-world

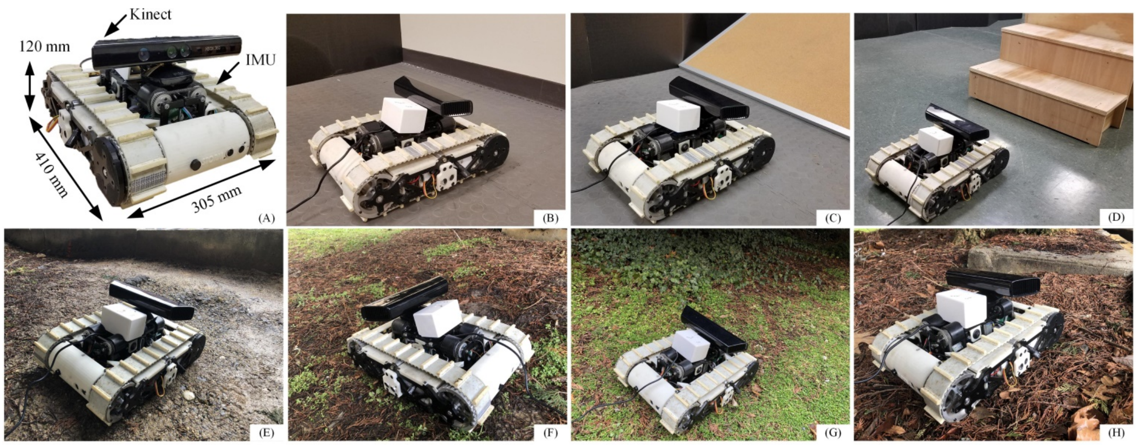

Fig. 4 Experimental validation in structured and unstructured terrains: (A) Experimental platform, (B) Facing a wall, (C) Facing a ramp, (D) Facing steps, (E) Facing a wall in an outdoor terrain, (F) Going downhill, (G) Going uphill at an angle, (H) Facing a step in an outdoor terrain.

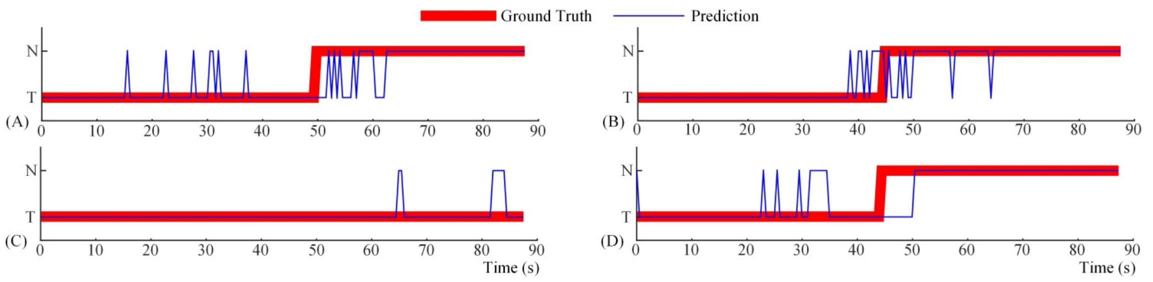

Fig. 5 Variation of traversability estimates over time in structured and unstructured terrains: (A) Facing a wall at an angle (Indoor), (B) Facing an edge of the platform (Indoor), (C) Going downhill (Outdoor), (D) Facing a wall (Outdoor). T denotes traversable and N denotes non-traversable.

scenario could be different from the math model underlying the physics simulation. This is particularly true in case of training using synthetic RGB images [21], [22]. Even though the trained architecture may work well in simulation, model mismatch issues can make transfer of knowledge from the physics simulator to the real-world difficult. As such, it is important to validate the performance of the trained architecture in real-world conditions.

The experimental validation involved testing the proposed NN architecture in five structured indoor and four unstructured outdoor scenarios, some of which are shown in Fig. 4. Fig. 4 (B)-(D) denote indoor conditions and (E)-(G) denotes outdoor conditions. Similar to the simulated scenario, a tracked robotic system, STORM, fitted with a Kinect sensor and IMU, was used for the experimental validation. As in the case of data collection simulations, a depth image of the terrain in front of the robot, along with the orientation of the robot was sent to the trained NN architecture. It should be noted that the trained architecture was deployed on the hardware for experimental validation without any additional tuning. Moreover, the outdoor experiments were conducted after sunset to reduce the infrared interference from sunlight with the Kinect.

TABLE I
EXPERIMENTAL VALIDATION RESULTS

Terrain condition	Roll (Degree)	Pitch (Degree)	Accuracy	Ground truth
Facing steps (I*)	-1.00	-0.88	91.43%	N*
Facing wall (I)	0.37	-2.25	88.57%	N
Facing wall at an angle (I)	0.74	-2.06	86.00%	N
Facing edge of platform (I)	-0.62	-1.44	89.71%	N
Facing ramp (I)	-0.25	-2.32	70.29%	T*
Facing wall (O*)	-2.64	-13.6	86.93%	N
Going downhill (O)	-5.80	28.88	96.00%	T
Going uphill (O)	-1.60	-35.70	92.00%	T
Facing step (O)	-1.36	10.81	85.71%	N

* I – Indoor, O – Outdoor, T – Traversable, N – None-Traversable.

In each case the robot was placed at a distance from possible obstacles and the low level controller on the robot was used to drive the robot straight with a constant linear velocity of 0.01 m/s. Throughout the process, the onboard sensor data was used to estimate traversability in real-time. For all the nine test cases, roll and pitch of the robot, and the accuracy as compared to ground truth are summarized in Table 1. The real-time traversability estimates from four different scenarios are shown in Fig. 5. For non-traversable cases the ground truth shifts from traversable to non-traversable when the robot is 1m away from the obstacle. As mentioned in [21], "Depth images can abstract away many of the challenging appearance properties of real-world objects". This helped in transitioning the neural network policy from simulated training to real world implementation without significant loss of performance as evidenced from Table 1 and Fig. 5. The results show that the proposed technique works well in both indoor and outdoor scenarios.

VI. PATH PLANNING APPLICATION

The feasibility of using real-time traversability predictions as obtained from the trained architecture for incremental path planning applications was validated through simulations. The same tracked robotic vehicle model used in the data collection simulations was also used for the path planning simulation.

The traversability estimations from the trained architecture were used to update an incremental planner, D* Lite [23] to enable an autonomous tracked vehicle to successfully navigate to a goal location without any prior maps. The overall navigational architecture consisted of D*Lite as the high-level planner which starts off with a grid map (each unit being 0.5x0.5 m) having the start and goal locations, the low level PID controller that drives the robot to each waypoint as dictated by the planner, and the NN architecture that informs the planner about possible obstacles in front of the robot. In order to reject false detections, the map was updated only when the NN predicted non-traversability continuously for five iterations. The planner then updates its map based on the detected obstacles. The simulation was performed in V-REP, with the navigation architecture implemented in python. For the duration of the simulation, the x, y position and roll, pitch, yaw orientations of the robot were given to the navigation architecture along with the depth images, which in turn returned the control inputs to V-REP. The path planning simulation was performed on unstructured terrain map (3.5x3.5 m), outside of the training data. Under the proposed navigation architecture, the robot avoids the steep hill in the middle and goes around as shown by the topography map in Fig. 6. It can be seen clearly that the robot successfully modifies its plan in real-time based on the traversability estimates during navigation. It should be noted that the robot did not possess any other sensors to detect obstacles

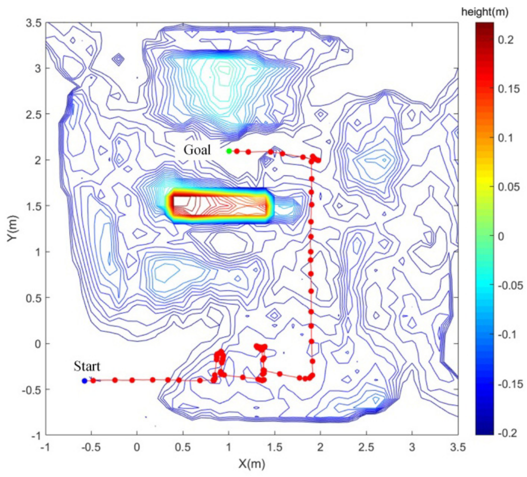

Fig. 6 Path planning in unstructured terrain condition. The start and goal locations are marked in blue and green dots, respectively. The path followed by the robot is marked in a solid red line.

and the path planning algorithm was solely informed by the trained NN architecture.

VII. Conclusion

This paper presented a neural network architecture that takes in depth images, along with roll and pitch angles of the robot to perform real-time traversability estimation. The training data was obtained from a robotic simulator that allowed for a wide variety of training conditions to be simulated, without causing any damage to the actual robot. The developed approach was validated in indoor and outdoor conditions, along with applications to incremental path planning. As mentioned in Section V, the experimental validation was limited due to the interference of Kinect data with sunlight. Even though this paper described the use of a Kinect to obtain the depth data, the overall approach is generalized towards the use of any depth sensor such as Velodyne, capable of working outdoors.

Future work will explore different architectures such as Long Short Term Memory to improve the performance of the learning based traversability estimation techniques. Color images of the environment as obtained from cameras could be easily integrated into the proposed architecture to improve the estimation accuracy. Most robotic simulators including V-REP and Gazebo have photorealistic rendering modes with accurate camera models that could be used to generate the training data in such cases. Similarly, slip experienced by the vehicle is another major factor that could affect the traversability of a given region. Including slip and color images to the architecture and analyzing their effect on traversability predictions will be explored as part of future work. Detailed performance evaluation of the proposed approach in comparison with existing probabilistic methods will also be addressed in the future.

References

[1] F. Fahimi, *Autonomous robots : modeling, path planning, and control*. Springer, 2009.

[2] C. Wong, E. Yang, X.-T. Yan, and D. Gu, “Adaptive and intelligent navigation of autonomous planetary rovers — A survey,” in *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2017, pp. 237–244.

[3] T. Shan, J. Wang, B. Englot, and K. Doherty, “Bayesian Generalized Kernel Inference for Terrain Traversability Mapping,” in *Proceedings of The 2nd Conference on Robot Learning*, 2018, vol. 87, pp. 829–838.

[4] T. M. Howard and A. Kelly, “Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots,” *Int. J. Rob. Res.*, vol. 26, no. 2, pp. 141–166, Feb. 2007.

[5] P. N. Currier, “A method for modeling and prediction of ground vehicle dynamics and stability in autonomous systems,” Doctoral Dissertation, Virginia Tech, 2011.

[6] B. Sebastian and P. Ben-Tzvi, “Physics Based Path Planning for Autonomous Tracked Vehicle in Challenging Terrain,” *J. Intell. Robot. Syst.*, vol. 95, no. 2, pp. 511-526, August 2019.

[7] P. Papadakis, “Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, Apr. 2013.

[8] L. Murphy, S. Martin, and P. Corke, “Creating and using probabilistic costmaps from vehicle experience,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4689–4694.

[9] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, “Learning Ground Traversability From Simulations,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1695–1702, Jul. 2018.

[10] A. Maligo and S. Lacroix, “Classification of Outdoor 3D Lidar Data Based on Unsupervised Gaussian Mixture Models,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 5–16, Jan. 2017.

[11] B. Suger, B. Steder, and W. Burgard, “Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-lidar data,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3941–3946.

[12] M. A. Bekhti, Y. Kobayashi, and K. Matsumura, “Terrain traversability analysis using multi-sensor data correlation by a mobile robot,” in *2014 IEEE/SICE International Symposium on System Integration*, 2014, pp. 615–620.

[13] S. Martin and P. Corke, “Long-term exploration & tours for energy constrained robots with online proprioceptive traversability estimation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5778–5785.

[14] J. Sock, J. Kim, J. Min, and K. Kwak, “Probabilistic traversability map generation using 3D-LIDAR and camera,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5631–5637.

[15] Coppelia Robotics, “V-REP,” 2010. [Online]. Available: http://www.coppeliarobotics.com/. [Accessed: 11-Feb-2017].

[16] Blender Foundation, “Blender v2.7.” [Online]. Available: https://www.blender.org/. [Accessed: 01-Jan-2017].

[17] P. Kumar, W. Saab, and P. Ben-Tzvi, “Design of a Multi-Directional Hybrid-Locomotion Modular Robot With Feedforward Stability Control,” in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), Cleveland, Ohio, Aug. 6-9, 2017.*

[18] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA*, 2013, vol. 28.

[19] M. Abadi *et al.*, “TensorFlow: a system for large-scale machine learning,” in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, 2016, p. 44.

[20] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*. Preprint ArXiv, 2014.

[21] K. Bousmalis *et al.*, “Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping.” Preprint ArXiv, 2018.

[22] J. Michels, A. Saxena, and A. Y. Ng, “High speed obstacle avoidance using monocular vision and reinforcement learning,” in *Proceedings of the 22nd international conference on Machine learning - ICML ’05*, 2005, pp. 593–600.

[23] S. Koenig and M. Likhachev, “D* Lite,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 476–483.