TYPE Original Research
PUBLISHED 28 June 2023
DOI 10.3389/frobt.2023.1086798

An embarrassingly simple approach for visual navigation of forest environments

Chaoyue Niu*, Callum Newlands, Klaus-Peter Zauner and Danesh Tarapore

School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom

OPEN ACCESS

EDITED BY
Junpei Zhong,
Hong Kong Polytechnic University, Hong Kong SAR, China

REVIEWED BY
João Filipe Ferreira,
Nottingham Trent University, United Kingdom
Bin Chen,
The University of Sheffield, United Kingdom

*CORRESPONDENCE
Chaoyue Niu,
✉ cn1n18@soton.ac.uk

RECEIVED 01 November 2022
ACCEPTED 14 June 2023
PUBLISHED 28 June 2023

CITATION
Niu C, Newlands C, Zauner K-P and Tarapore D (2023), An embarrassingly simple approach for visual navigation of forest environments.
Front. Robot. AI 10:1086798.
doi: 10.3389/frobt.2023.1086798

Navigation in forest environments is a challenging and open problem in the area of field robotics. Rovers in forest environments are required to infer the traversability of *a priori* unknown terrains, comprising a number of different types of compliant and rigid obstacles, under varying lighting and weather conditions. The challenges are further compounded for inexpensive small-sized (portable) rovers. While such rovers may be useful for collaboratively monitoring large tracts of forests as a swarm, with low environmental impact, their small-size affords them only a low viewpoint of their proximal terrain. Moreover, their limited view may frequently be partially occluded by compliant obstacles in close proximity such as shrubs and tall grass. Perhaps, consequently, most studies on off-road navigation typically use large-sized rovers equipped with expensive exteroceptive navigation sensors. We design a low-cost navigation system tailored for small-sized forest rovers. For navigation, a light-weight convolution neural network is used to predict depth images from RGB input images from a low-viewpoint monocular camera. Subsequently, a simple coarse-grained navigation algorithm aggregates the predicted depth information to steer our mobile platform towards open traversable areas in the forest while avoiding obstacles. In this study, the steering commands output from our navigation algorithm direct an operator pushing the mobile platform. Our navigation algorithm has been extensively tested in high-fidelity forest simulations and in field trials. Using no more than a 16 × 16 pixel depth prediction image from a 32 × 32 pixel RGB image, our algorithm running on a Raspberry Pi was able to successfully navigate a total of over 750 m of real-world forest terrain comprising shrubs, dense bushes, tall grass, fallen branches, fallen tree trunks, small ditches and mounds, and standing trees, under five different weather conditions and four different times of day. Furthermore, our algorithm exhibits robustness to changes in the mobile platform's camera pitch angle, motion blur, low lighting at dusk, and high-contrast lighting conditions.

KEYWORDS

low-viewpoint forest navigation, low-cost sensors, small-sized rovers, sparse swarms, depth prediction, compliant obstacles, forest simulation, off-road navigation

1 Introduction

The United Nations Global Forest 2021 sustainability study estimates forests to cover over 31% of landmass, an area of around 4 billion hectares (DESA, 2021). The management, maintenance and conservation of our forests is an enormous operation and of substantial importance to the economy and the environment. A sparse swarm of rovers could assist foresters in monitoring these ecosystems from the ground (Tarapore et al., 2020), gathering spatio-temporal environmental data across vast areas. For example, the swarm could gather census data on healthy tree saplings. It could visually inspect tree barks for symptoms of devastating invasive diseases, allowing identified trees to be precisely managed (Karpyshev et al., 2021). Importantly, to reduce their environmental impact, such as from soil compaction (Batey, 2009), and to allow their large-scale deployment as a swarm, the individual rovers have to be small-sized (portable) and inexpensive.

Off-trail navigation in forest environments remains a difficult task and an open problem in field robotics (Yang et al., 2018). Forest terrain consists of a variety of different vegetation including leaves, twigs, fallen branches, barks, stems, grass, bushes, and creeping vegetation. Rovers are required to predict their traversability over such *a priori* unknown terrains, relying solely on onboard sensors, under varying lighting and weather conditions (Niu et al., 2020; da Silva et al., 2021). Predicting rover-terrain interactions, is consequent not only to the innate characteristics of the forest terrain and the weather conditions (e.g., wet vs dry leaves), but also the dynamics of the interaction between the rover and the terrain which itself is susceptible to change (e.g., long grass tangled around axle of rover, thick layer of mud stuck on rover's wheels) (Ostafew et al., 2016). The problem is further compounded for small-scale rovers, such as that of a swarm: their small-size affords them only a limited low perspective of their surrounding terrain. Furthermore, consequent to their small size, almost all encountered vegetation is an obstacle, and a small rover is prone to toppling over at obstacles.

Many studies have investigated terrain traversability for navigation in off-road environments (reviewed in Papadakis (2013); Borges et al. (2022)), pioneered by the DARPA PerceptOR (Krotkov et al., 2007) and later the DARPA Learning Applied to Ground Vehicles (Jackel et al., 2006; Huang et al., 2009a; b) and Unmanned Ground Combat Vehicle Perceptor Integration programs (Bagnell et al., 2010; Silver et al., 2010). The majority of these studies discern geometry-based features of the terrain to predict traversability (Santana and Correia, 2011; Santamaria-Navarro et al., 2015; Tang et al., 2019; Haddeler et al., 2020; Lee and Chung, 2021; Wellhausen and Hutter, 2021). For example, Santamaria-Navarro et al. (2015) adopted a time-of-flight camera to determine obstacles online by thresholding the locally estimated normal orientation of queried planar patches of terrain. The authors also trained a Gaussian process model to classify traversable regions offline using terrain slope and roughness features from 3D point cloud data. Similarly, in Lee and Chung (2021), a traversability model was trained using slope, roughness, and curvature features, inferred from eigenvectors and eigenvalues of the covariance matrix of the terrain elevation map. In another example, a large-sized rover in simulation used 3D point cloud data from LiDAR sensors to estimate the gradient of uneven terrain, and consequently quantify the mechanical effort in traversing the terrain Lourenço et al. (2020); Carvalho et al. (2022). Importantly, while geometry-based approaches for terrain traversability have demonstrated some success in navigating rigid terrains such as on well paved paths in structured urban environments (e.g., Bellone et al., 2017; Tang et al., 2019; Liu L. et al., 2020; Lee and Chung, 2021; Lee et al., 2022), they may face potential challenges on compliant terrains such as a forest floor, at a low-viewpoint, with an abundance of grass and other soft vegetation where geometry-based features are unreliable (Haddeler et al., 2022). In such environments, these approaches would potentially result in incomplete elevation maps due to the limitations of the depth sensor hardware.

Geometry-based exteroceptive information are often coupled with proprioceptive information to improve the robustness of terrain analysis (Borges et al., 2022). Here, data-driven near-to-far learning approaches are typically employed to correlate geometry-based features with proprioceptive features such as the rover's attitude (Murphy et al., 2012; Ho et al., 2013a; Bjelonic et al., 2018; Wolf et al., 2018; Haddeler et al., 2020). The resultant mobility prediction model is commonly used with optimization techniques, such as dynamic programming, to select actions that maximize stability (e.g., see Peynot et al. (2014)). In other studies, geometry-based features are coupled with appearance-based visual features for terrain segmentation and classification (Milella et al., 2015; Schilling et al., 2017; Kragh and Underwood, 2020; Chen et al., 2022; Haddeler et al., 2022). For example, Milella et al. (2015) augmented terrain geometry information from a short-range radar sensor with color and texture information from a long-range monocular camera. The authors used terrain slope information to automatically label traversable and untraversable areas in close-range, that consequently serve as training labels for a long-range visual classifier. In effect, using near-to-far learning, the short-range narrow field-of-view of the radar sensor is extended to the long-range wide field-of-view of a monocular camera.

The existing approaches to off-road navigation appear unsuitable for *inexpensive small-sized forest rovers*. Large-sized rovers, typically used in studies for off-road navigation, perceive the environment from a high viewpoint of around 1 m (e.g., Ho et al. (2013a; b); Ugenti et al. (2022); Fnadi et al. (2020); Baril et al. (2022); Bagnell et al. (2010), also see comparison in Table 1 and Figure 1A). In comparison, for small-sized forest rovers, discerning the forest scene from a size-proportional low viewpoint on the order of centimeters is relatively difficult due to the limited field-of-view in the vertical direction. The field-of-view will frequently be partially occluded by compliant obstacles in proximity, such as grass, leaves, and low-hanging branches. In such scenarios, with large number of proximal compliant obstacles, a fine-grained analysis of the terrain to discern relevant geometric-features may be a waste of computing power. Additionally, geometry-based approaches for traversability analysis commonly use LiDAR or other expensive depth sensors (see Table 1 and Figure 1B), which are not scalable for deployment on large-scale swarms of rovers. Moreover, with appearance-based approaches, environmental factors such as shadows and high-contrast lighting severely affect the visual appearance of the terrain and consequently place high computational demands for feature discrimination (Corke et al., 2013; Ai et al., 2022). Reliable and robust feature discrimination is also sensitive to motion blur, which is more prominent in small-sized rovers due to the unintended tilting and rolling from moving on uneven terrain.

We propose the design of a low-cost navigation system for small-sized forest rovers, solely using low-resolution depth-prediction images. Our research contribution lies in the uncharted bottom-right region in Figure 1. In our study, a light-weight convolution neural network is used to predict the depth image for the rover from an RGB input image from a monocular camera. A

TABLE 1 A comparison of terrain traversability studies in off-road environments, with the weight of the rover and the approximate cost of the sensors required for navigation. Sensor costs were obtained from vendor sites, where available. Dashed lines indicate the corresponding data was unavailable.

Index	Reference	Approximate sensors cost (GBP)	Weight of rovers (kg)
1	Ostafew et al. (2016)	3000	900
2	Peynot et al. (2014)	12000	30
3	Ho et al. (2013a), Ho et al. (2013b)	12000	30
4	Bjelonic et al. (2018)	5700	10
5	Krebs et al. (2010)	1600	35
6	Palazzo et al. (2020)	5400	-
7	Murphy et al. (2012)	3800	3
8	Santamaria-Navarro et al. (2015)	800	108
9	Milella et al. (2015)	2300	-
10	Gerdes et al. (2020)	5200	310
11	Mayuku et al. (2021)	2600	50
12	Wolf et al. (2018)	3000	740
13	Kragh and Underwood (2020)	60000	108
14	Ugenti et al. (2022)	2300	206
15	Tang et al. (2019)	4100	20
16	Lee and Chung (2021)	13000	-
17	Sebastian et al. (2019)	122	9
18	Haddeler et al. (2020)	4000	92
19	Wellhausen and Hutter (2021)	400	30
20	Ref in *cf.* Jackel et al. (2006); Huang et al. (2009a,Huang et al. (2009b)	-	109
21	Hoeller et al. (2013)	-	375
22	Liu et al. (2020a)	600	200
23	Braun et al. (2009)	-	750
24	González et al. (2009); Gonzalez et al. (2013)	200	500
25	Fnadi et al. (2020)	35000	880
26	Chen et al. (2022)	4800	62
27	Schilling et al. (2017)	7000	50
28	Baril et al. (2022)	7600	590
29	Howard et al. (2008), Howard et al. (2010); Bagnell et al. (2010); Silver et al. (2010)	-	6000

simple coarse-grained navigation algorithm is devised to steer the rover towards open traversable areas in the forest, using mean depth information. Due to additional challenges in designing a high endurance locomotion system for a small-sized low-cost rover, in this study we focus solely on the navigation system. Therefore, our mobile platform is pushed manually by an operator, guided by steering actions on an onboard display. Our developed low-viewpoint navigation algorithm is robust to changes in the camera pitch angle, motion blur, high-contrast lighting, and low-lighting at dusk conditions. It uses a low-resolution 16 × 16 pixel depth prediction image from a 32 × 32 pixel monocular RGB image, and runs on a Raspberry Pi 4. Utilizing low-resolution input images reduces the computational and energy requirements for the rover, enabling efficient navigation for platforms with limited computing capabilities. Our algorithm has been extensively tested in high-fidelity forest simulations and in field trials, successfully navigating a total of over 750 m of real-world forest terrain under five different weather conditions and four different times of day.

2 Materials and methods

Our algorithm for forest navigation comprises of two steps: i) A prediction model is used to infer the depth image from a low-viewpoint low-resolution RGB image input from a monocular camera mounted on our mobile platform; and ii) The output depth

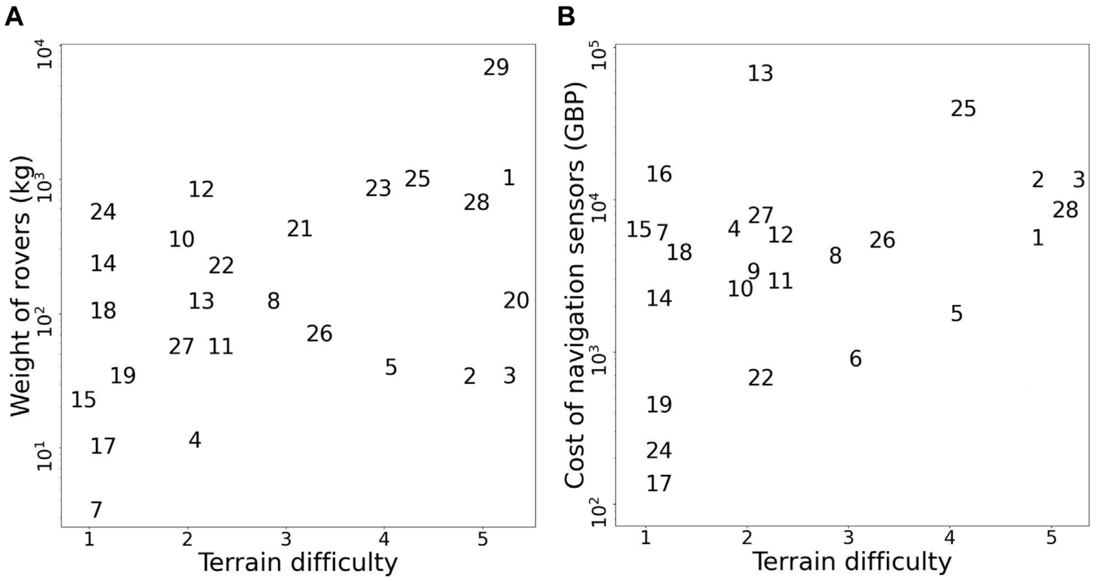

FIGURE 1
The relationship between terrain difficulty and navigation sensor cost **(A)**, and the weight of the rover **(B)**. Terrains are categorized in ascending order of difficulty as follows: 1. Paved road, short grass; 2. Sandy soil, paved trail; 3. Sparse bushes, tall grass; 4. Rubble, dense bushes, and gravel; and 5. Mars-analogue environment, forest environment–dense bushes, tall grass, fallen branches, fallen tree trunks, standing trees, small mounds and ditches. Importantly, to the best of our knowledge, references 1, 28 and 29 are the only studies investigating navigation in forest environments. All studies referenced are indexed in Table 1.

image is utilised by our steering algorithm to provide direction motion commands to the operator pushing the mobile platform, to navigate it towards the goal while avoiding obstacles. An overview of our approach for forest navigation is illustrated in Figure 2.

2.1 Depth prediction model

Inferring depth from low-resolution RGB images will allow us to use geometric information for navigation, while avoiding the high cost of depth sensors. The depth prediction model has to infer depth in real-time, on an embedded computer, with sufficient accuracy to facilitate navigation of the forest environment. We propose to use a Densenet deep neural network (Huang et al., 2017) for depth prediction, using Densenet as an encoder and following an auto-encoder architecture through transfer learning. In previous studies, Densenet models–Densenet-169 have been successfully used for monocular depth prediction (e.g., see Alhashim and Wonka (2018) trained on both indoor (Nathan Silberman et al., 2012) and outdoor autonomous driving (Geiger et al., 2013) datasets). To the best of our knowledge, no previous studies have validated monocular depth prediction models on unstructured forest datasets.

For our study, the DenseNet encoder pre-trained on the ImageNet dataset (Krizhevsky et al., 2017) was employed. The DenseNet-121 auto-encoder network was then trained on our real-world low-viewpoint forest dataset, comprising a total of 160,000 RGB-D image pairs at 640 × 480 resolution, with ground truth depth in range [0.2 m, 10 m] (Niu et al., 2020). Hyperparameters for training were selected by trial and error using a subset of this dataset, consisting of 5,500 (training), 3,800 (validation), and 1,000 (testing) RGB-D image pairs. During hyperparameter selection, we trained three DenseNet networks–DenseNet-121, DenseNet-169, and DenseNet-201. Consequently, considering the runtime on a Raspberry Pi 4, accuracy, and model size, the Densenet-121, trained for 20 epochs, was selected. The selected model was trained on our entire forest dataset, using 140,000, 10,000, and 8,450 RGB-D image pairs for training, validation and testing, respectively. We ensured that the images in the test set were of an entirely different segment of the forest, compared to the training and validation sets.

The training was implemented using the TensorFlow Keras (Abadi et al., 2016) library on a NVIDIA GTX 1080ti (11G) GPU. An Adam optimizer (Kingma and Ba, 2014) was applied for loss minimization with the default parameters of $B_1 = 0.9, B_2 = 0.999$, learning rate $\lambda = 0.001$ and batch size of 8. The loss function was composed of the following: i) Point-wise L1 loss; ii) L1 loss over the image gradient; and iii) Structural similarity loss (for details see Alhashim and Wonka (2018)). During training, data augmentation was applied to reduce over-fitting, including geometric mirroring of RGB images with a probability of 0.5, and photometric swapping of G and R channels from the input RGB images with a probability of 0.5. Note that with our DenseNet-121 model architecture, the resolution of the output predicted depth image was half that of the input RGB image in both dimensions.

We converted the trained Tensorflow model into a TensorFlow-Lite model (Louis et al., 2019), resulting in a three-fold improvement in run-time performance. In estimating the required run-time for our mobile platform, we assume the average speed of a small-sized rover navigating over forest terrain as 0.2 m/s. Additionally, considering the minimum depth range of our ground truth is 0.2 m (Niu et al., 2020), the depth prediction model implemented

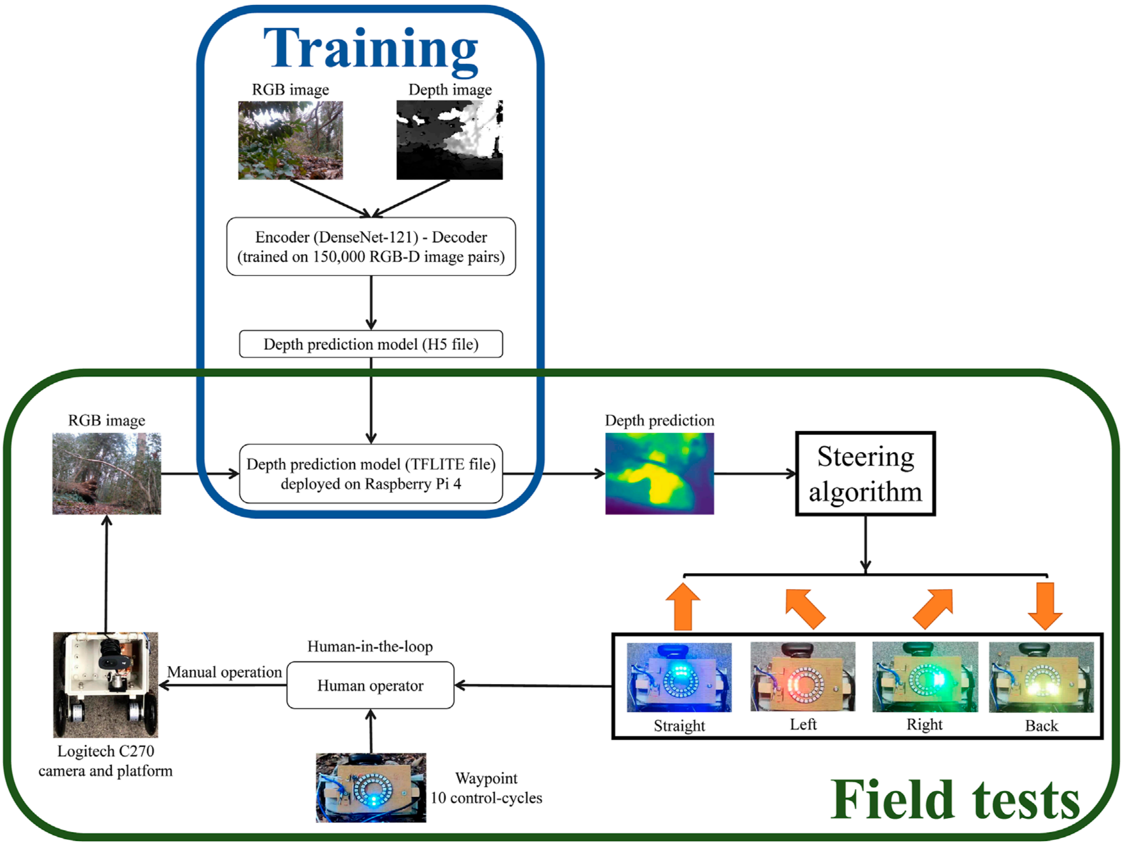
Training
RGB image
Depth image
Encoder (DenseNet-121) - Decoder (trained on 150,000 RGB-D image pairs)
Depth prediction model (H5 file)
RGB image
Depth prediction model (TFLITE file) deployed on Raspberry Pi 4
Depth prediction
Steering algorithm
Straight
Left
Right
Back
Human-in-the-loop
Human operator
Manual operation
Logitech C270 camera and platform
Waypoint 10 control-cycles
Field tests

FIGURE 2
Overview of our approach to forest navigation. During the model training stage, a monocular depth prediction model is trained using RGB-Depth image pairs obtained from the Intel Realsense D435i camera. The model architecture employs an encoder-decoder setup, with the encoder utilizing the DenseNet-121 network and the decoder employing standard convolutional layers. For the subsequent field trials conducted in real-world forest environments, the trained model was converted from an H5 file to a TFLITE file for faster inference. This TFLITE file was then deployed on a Raspberry Pi 4. The deployed model takes input from a Logitech C270 webcam, processes the image, and generates steering actions using a coarse-grained steering algorithm. These steering actions are displayed on LED boards. To execute the generated steering instructions, a human operator interacts with the platform. The operator pushes the platform, following a fixed go-forward displacement and turning angle, and aligns it with the goal waypoint at a fixed interval of every 10 control cycles. The approach combines model training using RGB-Depth images, model deployment on a Raspberry Pi and human-in-the-loop with LED instructions.

on an embedded computer with a runtime of about 1 s may be sufficient to achieve real-time performance. Our original 640 × 480 images resulted in a runtime of around 12 s per image on a Raspberry Pi 4. To identify the necessary downscaling, runtimes were tested across several RGB input image resolutions (see Table 2). Since the resolution of 32 × 32 met our real-time requirement, we downsampled the RGB image resolution of our real-world forest dataset from 640 × 480 to 32 × 32, prior to training our Densenet-121 depth prediction model using the same setup. The 32 × 32 Tensorflow-Lite model is 42 MB in size and has a runtime of 0.80 ± 0.02s (mean ± SD) on a Raspberry Pi 4.

We investigate the quality of depth prediction for navigation with high and low resolution input RGB images. With high resolution input RGB images (640 × 480), obstacles in the foreground and background are clearly distinguishable (see example in Figure 3) Despite employing low-resolution RGB images, qualitative results indicate that the depth prediction model can still distinguish between obstacles in the foreground and background (Figure 4). The model appears to provide reasonable predictions in high-contrast scenes (see Figure 4B, the bright and shadowed forest ground in close vicinity have similar predicted depths). As one would expect with the low resolution employed, the results are little affected by motion blur (Figure 4C). Furthermore, obstacles in the foreground and background remain distinguishable in low lighting conditions such as at dusk (see uprooted tree trunk in Figure 4D).

2.2 Steering algorithm

Our steering algorithm receives a predicted depth image at every control-cycle, and indicates one of four possible steering directions to the human operator pushing the robot *via* an onboard LED display. The steering directions are “Go-straight”, “Turn-left”, “Turn-right”, “Go-back”, and (orientate towards) “Waypoint”. With the “Go-straight” action the platform is moved straight approximately 50 cm forward. Rotatory actions of “Turn-left” and “Turn-right” pivot the platform by approximately 15° along the yaw axis. Similarly, the “Go-back” action rotates the platform by approximately 180°. Finally, with the “Waypoint” action the platform is rotated towards the

TABLE 2 Mean ± SD runtime of depth prediction with the DenseNet-121 model, aggregated across 100 randomly selected images, for different input RGB image resolutions. The depth prediction model was executed on a Raspberry Pi 4 (4 GB RAM).

Image resolution	32 × 32	128 × 96	128 × 128	256 × 192	256 × 256	640 × 480
Runtime	0.80 ± 0.02 s	1.35 ± 0.02 s	1.66 ± 0.06 s	2.88 ± 0.02 s	3.60 ± 0.16 s	11.94 ± 0.72 s

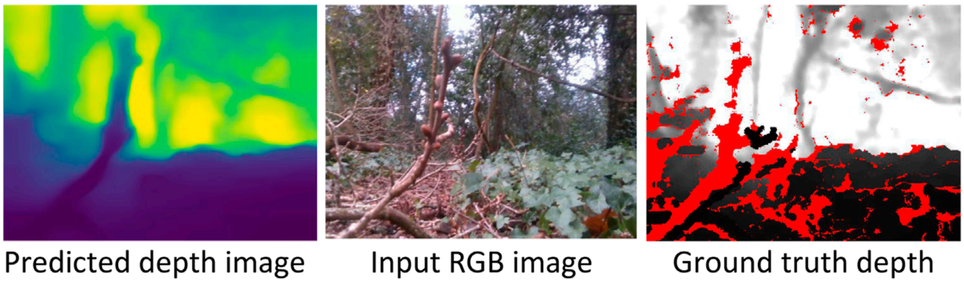

FIGURE 3
Sample 320 × 240 predicted depth image of a low-viewpoint forest scene from a 640 × 480 input RGB image. Darker (lighter) pixels in the depth image are closer (further away) from the camera. In the predicted depth image, the tree branch in the foreground is distinguishable from the standing trees in the background. The displayed predicted depth image has been upsampled here by a factor of two for visual clarity. Red pixels in the ground truth depth indicate that the depth could not be determined by the Intel RealSense D435i camera.

direction of the goal waypoint; this occurs here every 10 control-cycles and in general could be based on GPS information.

For steering, a simple algorithm is used for obstacle avoidance, to select the appropriate action from “Go-straight”, “Turn-left” and “Turn-right” (see example in Figure 5). The predicted depth image is first divided into three equal-sized vertical segments, each corresponding to the “Turn-left”, “Go-straight”, and “Turn-right” actions. Average depth across all the pixels in each column of the depth image is then calculated. The robot is directed to move in the direction associated to the segment with the highest mean depth value. A higher average depth value indicates a region with none or fewer obstacles, corresponding to a relatively more traversable direction. Additionally, averaging over the entire vertical segment allows the steering algorithm to be robust to inadvertent changes in the camera pitch angle during motion.

The “Go-back” direction is used to avoid collisions with close-range obstacles or to avoid encountering large untraversable areas (e.g., a fallen tree trunk) in the distance. It is triggered when the mean lower half of the predicted depth image is less than a predefined threshold of 0.7 m, the LED display then warns the operator of a potential collision. However, false-positives may occur, such as when the robot is on an incline, or if the camera is pointing down towards the ground. Therefore, following the first warning, the operator rotates the robot along the pitch axis by approximately 5°, so that the camera tilts upwards. If the collision warning continues to be displayed, the “Go-back” action is executed. To avoid potential collisions, the above procedure for the “Go-back” action takes priority over all the other steering actions.

2.3 Mobile platform

The mobile platform for our real-world forest experiments (see Figure 6) consists of a telescopic extension pole (1.21 m) and a CamdenBoss X8 series enclosure (L × W × H: 185 × 135 × 100 mm). A Logitech C270 HD webcam (diagonal 55° field of view) is mounted inside the enclosure at 20 cm above the ground and connected to a Raspberry Pi 4. Two black polyurethane scooter wheels, 100 mm in diameter and 24 mm in width, were mounted on the left and right sides underneath the enclosure, to facilitate traversal over rough terrain. A stripboard (95 × 127 mm) was fixed to two rectangular wooden blocks on the top of the enclosure, alongside two concentric NeoPixel rings of addressable RGB LEDs (Adafruit Industries, NY). The two NeoPixel rings were connected to the Raspberry Pi 4 (4 GB RAM) via a twisted pair (data) and a USB cable (power), and a Schmitt-trigger buffer (74LVC1G17 from Diodes Incorporated, TX) in the serial data line was used to overcome the capacitance of the long twisted pair wire. A HERO 9 (GoPro, CA) action camera was also mounted on the telescopic pole 50 cm from the top of the enclosure to allow for third-person view high-resolution video recording of the experiments. The overall cost of our mobile platform was around 250 GBP, with the sensor hardware for navigation—-a Logitech C270 camera and Raspberry Pi 4 embedded computer—costing a total of 70 GBP.

RGB images captured by the Logitech camera every 4 seconds–one control-cycle–were input to the depth prediction model deployed on the Raspberry Pi. Subsequently, the steering algorithm was applied to the predicted depth image, to output a steering direction, which was displayed on the NeoPixel rings (see Figure 6 for details on direction indications). Following the direction displayed, the operator directed the mobile platform to perform the required motion.

3 Experiments

Due to the challenging nature of forest field experiments, our navigation algorithm was first tested extensively in simulations (Section 3.1), before investigating its performance on-trail and off-trail in real-world forests (Section 3.2).

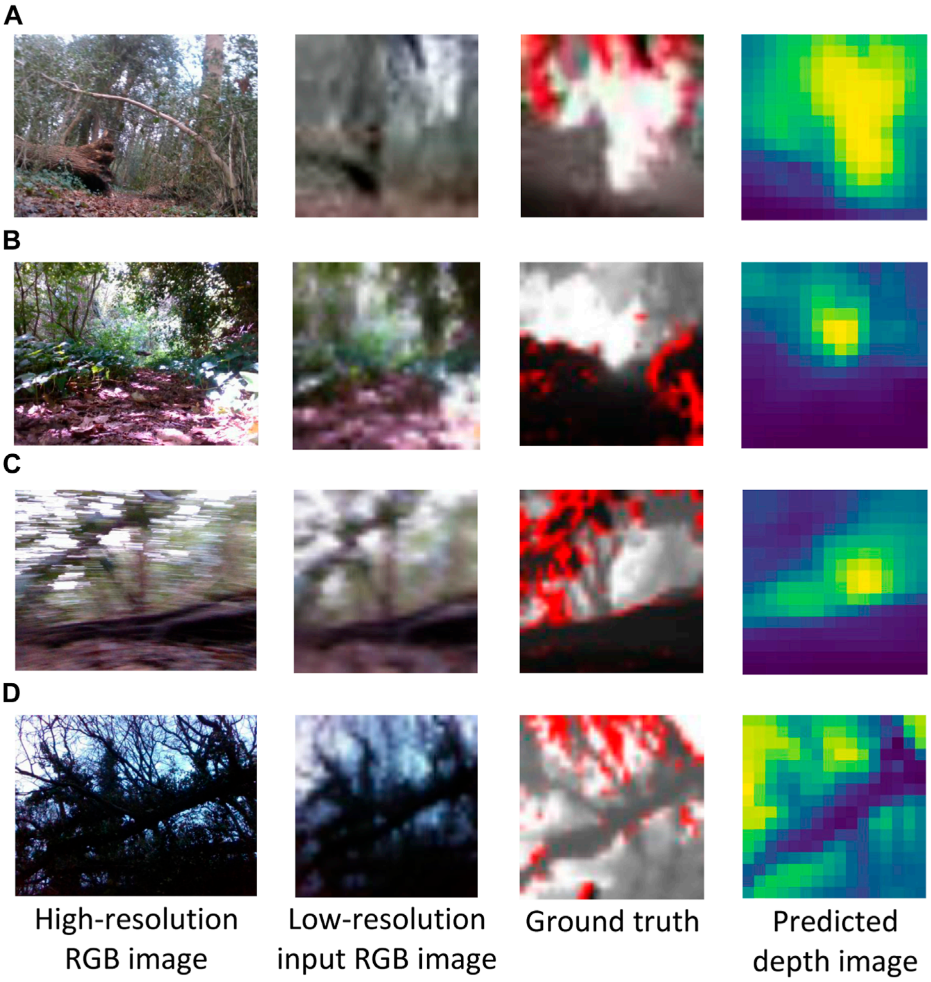

FIGURE 4
Sample low-resolution 16 × 16 depth images predicted from 32 × 32 input RGB images. The corresponding high resolution (640 × 480) RGB images are shown for context. The depth prediction model was assessed in cloudy **(A)**, high-contrast lighting **(B)**, motion blur **(C)** and low-lighting conditions at dusk **(D)**. RGB and ground truth depth images were captured with an Intel RealSense D435i camera.

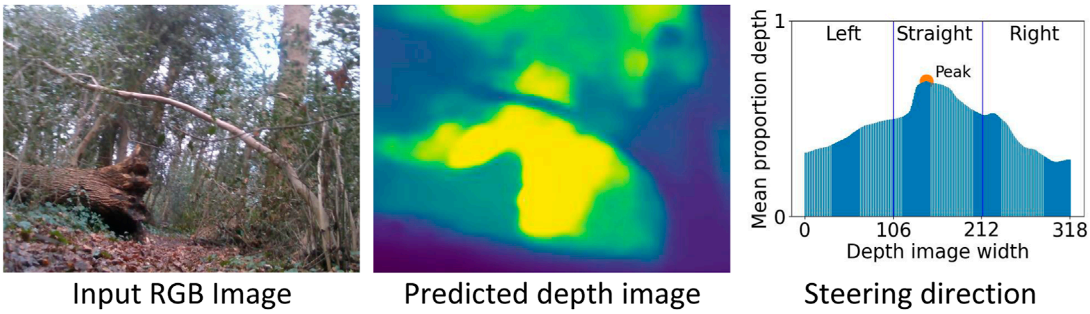

FIGURE 5
An example steering direction for an input RGB image (640 × 480), the corresponding predicted depth image (320 × 240), and the mean depth averaged across each column of the predicted depth image. As the peak mean depth lies in the central segment of the depth image, the mobile platform is directed straight (towards the traversable region between the fallen tree trunk and the standing tree). The displayed predicted depth image has been upsampled by a factor of two for visual clarity.

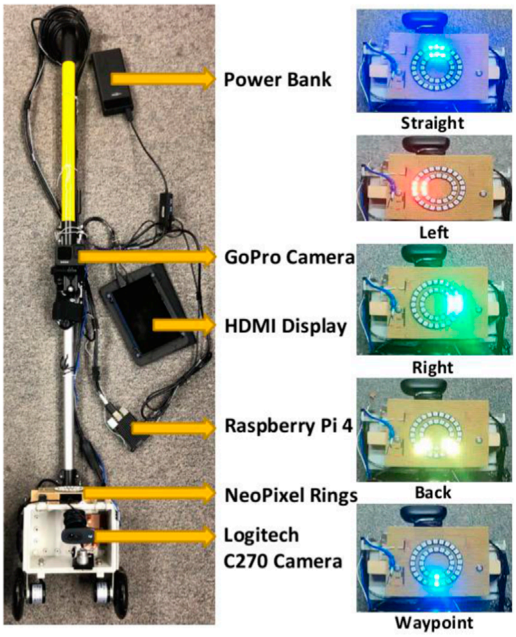

FIGURE 6
The two-wheeled mobile platform is equipped with a Logitech C270 camera, two NeoPixel rings, a Raspberry Pi 4, a Raspberry Pi HDMI display, a GoPro camera, and a portable power bank. RGB images captured by the Logitech camera are transmitted to the Raspberry Pi 4, to predict depth images for the steering algorithm. Resulting steering actions are displayed on the NeoPixel LED rings.

3.1 Navigating a simulated forest

A high-fidelity forest simulator, ForestGenerator (Newlands and Zauner, 2022), was used for the simulation experiments. ForestGenerator is a generalised, open-source tool for generating and rendering interactive, realistic forest scenes using a specialised L-systems, a custom ecosystem simulation algorithm and an OpenGL-based render pipeline and can be controlled from the command line (see Figure 7 for some example rendered scenes). Our simulated rover had the same dimensions as our physical platform. The simulation experiments were performed to investigate the capability of our steering algorithm to navigate the forest environment using low-resolution predicted depth images, in particular discerning how low can we go.

As with our field experiments, for experiments in simulation, a DenseNet-121 depth prediction model was used. The model was trained on a dataset of RGB-D image pairs generated using the ForestGenerator. The scenes comprised different tree species with a cumulative density of around 1 tree/16 m^2 (see above canopy view in Figure 7). For these images, the simulated camera had a maximum depth range of 10 m. It was positioned at pitch angles of 0° and 30°. For each pitch angle, images were generated from three different viewpoints (0.3 m, 0.5 m and 1 m), each under three different lighting conditions (RGB luminance of 87 ± 22, 99 ± 16 and 118 ± 10 respectively, Mean ± SD across 1000 images). For depth prediction, the DenseNet-121 auto-encoder network pre-trained on the ImageNet dataset was further trained over 20 epochs using 36,000 (training), 2,000 (validation) and 750 (testing) simulated RGB-D image pairs.

Our experiments in simulation were performed in a forest of size 50 × 50 m^2, with start and goal waypoints at (5 m, 5 m) and (45 m, 45 m) respectively, and with around 150 randomly distributed standing trees. Actions “Go-straight”, “Turn-left”, “Turn-right”, and “Waypoint” (defined in Section 2) were used to navigate the mobile platform towards the goal. For the experiments, the simulated camera was positioned at a low-viewpoint of 30 cm with a pitch angle of 0°, other parameters of the camera were the same as used in the synthetic dataset creation. We assessed the performance of the navigation algorithm using ground truth depth images and predicted depth images in separate experiments for each of six depth image resolutions of 16 × 16, 64 × 48, 64 × 64, 128 × 96, 128 × 128 and 320 × 240, replicated twenty times for each resolution. Therefore, in total 2 sensor-models × 6 image resolutions × 20 replicates = 240 experiments were performed in simulation.

Performance was assessed with the following metrics: i) Total distance traversed by the mobile platform to reach the goal waypoint; ii) The turning rate—the ratio of the number of “Turn-left” and “Turn-right” steering actions over the total number of actions—is zero for a straight-line trajectory and in general unbounded (arbitrary long detours and arbitrary many turns without forward progress); and iii) The number of collisions sustained. In case of a collision, the platform was repositioned next to the obstacle.

In all simulation experiments, the mobile platform was able to arrive at the goal waypoint. With the ground truth depth image, the steering algorithm was able to efficiently navigate the platform irrespective of the input image resolution, traversing a mean distance of 59.95 ± 1.2 m to reach the goal (across all six input image resolutions, see Figure 8A); the straight-line distance between start and goal is 56.57 m. The performance of the steering algorithm was almost constant across several different resolutions of the predicted depth image 64 × 48, 64 × 64, 128 × 96, 128 × 128 and 320 × 240 (mean traversed distance 59.45 ± 0.7 m). However, with the lowest resolution predicted depth images of 16 × 16 steering was less efficient, with the mobile platform traversing a distance of 63.68 ± 4.4 m to reach the goal.

Similar trends in performance were observed for the turning rate of the mobile platform (see Figure 8B). With the ground truth depth images, the turning rate was not influenced by the resolution, at 0.75 ± 0.02 across all six input image resolutions. The performance deteriorated to 0.85 ± 0.02 with predicted depth images.

The slightly poor performance in the distance traversed and the turning rate for low-resolution predicted depth images is ascribed to the coarse-grained prediction of depth of background obstacles. When obstacles in the foreground appeared few to none, this results in a higher uncertainty of steering action. Unable to discern narrow traversable gaps between trees in the background, the mobile platform may be directed to a detour around instead of in between the trees. In such scenarios, a potential solution may be to weigh the inferred steering direction towards the direction of the waypoint. Importantly, in navigating the simulated forest environment, the mobile platform sustained no collisions in almost all the replicates. In particular, at a 16 × 16 resolution, a single collision was sustained in each of three replicates, with no collisions sustained in the remaining 17 replicates (see Supplementary Table S1 in Supplementary Material).

FIGURE 7
The forest canopy and low-viewpoint RGB images of standing trees and other vegetation from an example forest synthesized by the ForestGenerator (Newlands and Zauner, 2022).

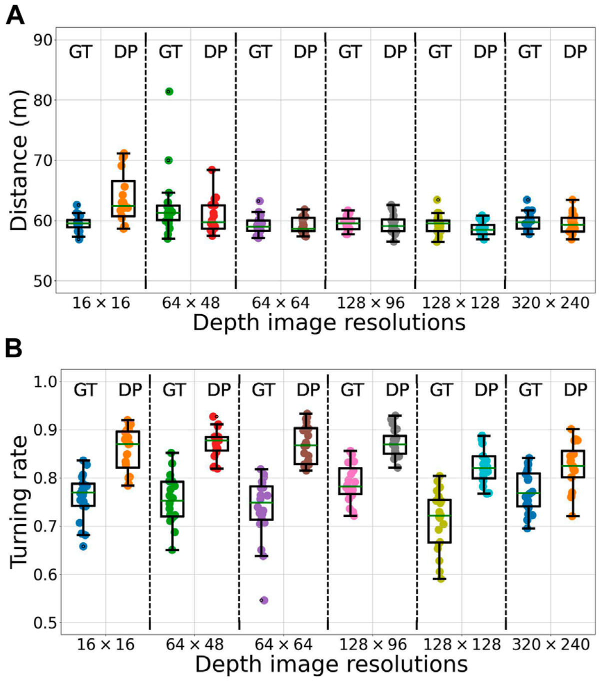

FIGURE 8
The distance traversed **(A)** and the turning rate **(B)** from start-to-goal for the mobile platform in simulated forests, across 20 replicates, in each of the 12 combinations of sensor-models—ground truth (GT) and predicted depth (DP)—and depth image resolutions—16 × 16, 64 × 48, 64 × 64, 128 × 96, 128 × 128 and 320 × 240. On each box, the mid-line marks the median, and the box extends from the lower to upper quartile below and above the median. Whisker outside the box indicate the maximum and minimum values, except in case of outliers, which are shown as dots. Outliers are data points outside of 1.5 times the interquartile range. Note that the *y*-axes do not start at zero.

In summary, rather than requiring a precise depth value for each pixel, our steering algorithm relies only on whether the foreground and nearby obstacles are visually distinguishable from the background. As such, our navigation algorithm is largely tolerant to inaccuracies in depth prediction. The somewhat lower performance of the navigation algorithm with a 16 × 16 resolution predicted depth image is largely offset by the high run-time performance, thus supporting its use for our real-world forest experiments.

3.2 Field experiments

Our field experiments were performed in the Southampton Common (Hampshire, UK), a woodland area of approximately 1.48 km^2. Experiments were performed in the following two sites of the Common: i) Following a long forest trail; and ii) Steering through a smaller but more challenging off-trail forest environment.

For the field experiments, besides the trajectory length, turning rate and number of collisions sustained, the following additional metrics were used to assess the performance of the navigation algorithm in navigating from start to goal waypoints: i) Time taken to reach the goal; ii) Number of true-positive incidents—the “Go-back” action is accurately triggered on encountering a large obstacle (e.g., fallen tree trunk) blocking the path of the mobile platform; iii) Number of false-positive incidents—the “Go-back” action is unnecessarily triggered, i.e., there are no obstacles obstructing the platform; and iv) Number of false-negative incidents—the “Go-back” action is not triggered in the presence of a large obstacle obstructing the platform, thus risking a potential collision. In total, seven metrics were used to assess navigation performance in our field experiments.Following a long forest trail. The mobile platform was tested on a dried mud trail comprising various forest obstacles. Obstacles included dense bushes, tall grass, leaf litter, fallen branches, fallen tree trunks, standing trees and a ditch formed at the roots of a large uprooted tree (see examples in Figure 9A).

For our experiments, the start and goal waypoints were positioned at (5056.2141 N, 124.0516 W) and (5056.1859 N, 124.1515 W), respectively (see Figure 10). The waypoints were selected to encompass a high diversity of compliant and rigid forest obstacles such as leaf litter, twigs, fallen branches, fallen tree trunks, standing trees, grass, bushes, and creeping vegetation (see examples in Figure 9A). The actions “Go-straight”, “Turn-left”, “Turn-right”, “Go-back” and “Waypoint” (defined in Section 2) were used to navigate the mobile platform towards the goal waypoint. As the goal was 210° SW of the start location, this bearing was used to rotate

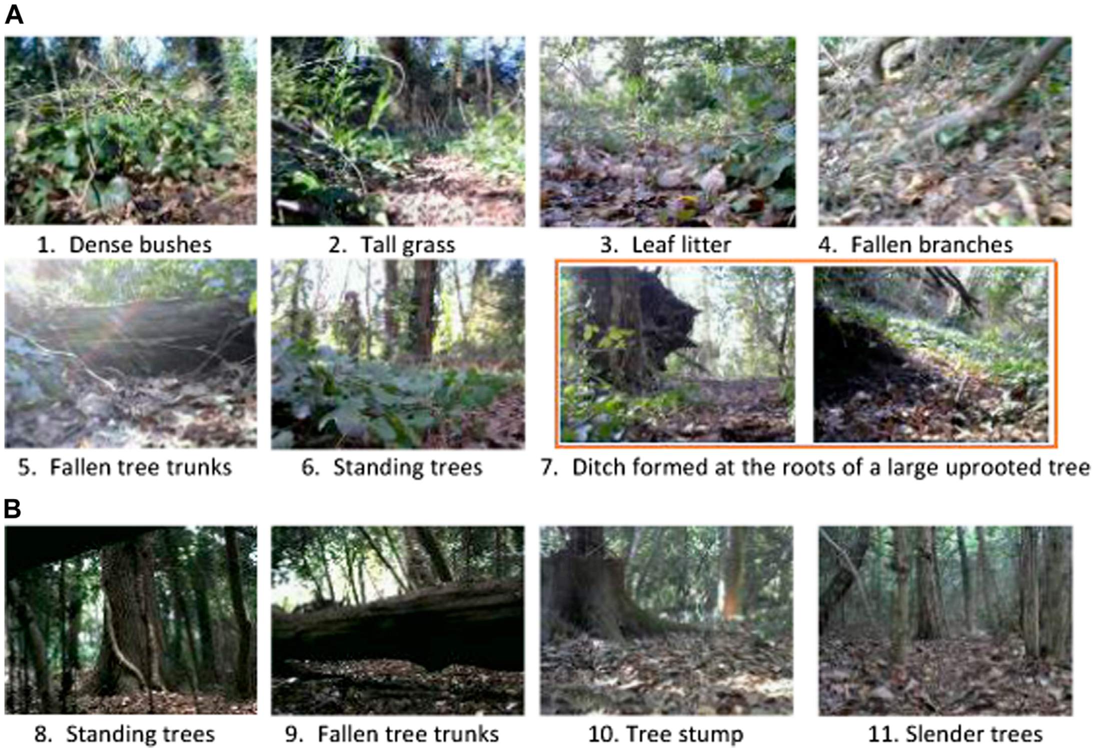

FIGURE 9
Examples of obstacles encountered by the mobile platform both on the forest trail **(A)** and off-trail **(B)** in the Southampton Common woodland.

the mobile platform to face the goal when the "Waypoint" action was triggered. The "Go-back" action was employed by the mobile platform to turn around and attempt to find an alternative path to circumvent large obstacles such as fallen tree trunks. If this action was triggered five times consecutively for the same obstacle, we assumed that there were no traversable paths around the obstacle; consequently, the operator would lift the platform over the obstacle, log the incident, and continue the experiment.

Trail experiments were performed five times in the forenoon, midday, and afternoon under weather conditions of cloudy, scattered clouds, mostly clear, and sunny (see details in Supplementary Table S2 of the Supplementary Material). Across all experiments, the platform was able to reach the goal waypoint, traversing a mean distance of 146 ± 3 m with a turning rate of 0.53 ± 0.05 in 20.7 ± 4.9 min (see Table 3). Our algorithm was largely able to steer the platform towards open spaces to avoid potential collisions (see examples in Figures 11A,B; for additional examples of steering by low-hanging tree branches, tall grass, and fallen tree trunks see Supplementary Figure S1 and the demonstration video of the Supplementary Material). In scenarios where the robot was facing a close-range obstacle, or large untraversable areas in the distance, the "Go-back" action was successfully triggered to avoid potential collisions (see Figure 11C—a fallen tree trunk covered in weeds and moss,; Figure 11D—dense bushes). The "Go-back" action was unnecessarily triggered only once, i.e., a false-positive incident, when the platform was facing uphill in a small ditch. Importantly, across all five experiments, the robot navigated the forest trail without sustaining any collisions, or incurring any false-negative incidents. Finally, once in each of the five experiments, due to a large fallen tree blocking the forest trail, the platform had to be lifted over it, as no traversable paths were found to circumvent the obstacle.

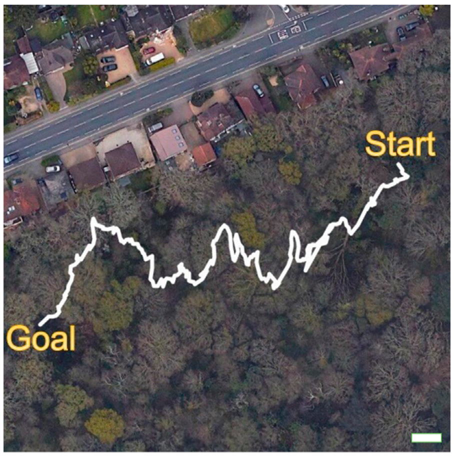

FIGURE 10
Trajectory from GPS metadata of the forest trail overlaid on an aerial view of the Southampton Common woodland. The white scale bar in the lower right corner corresponds to a distance of 10 m. Permitted use: Imagery © 2022 Getmapping plc, Infoterra Ltd & Bluesky, Maxar Technologies, The GeoInformation Group, Map data © 2022 Google.

TABLE 3 The trajectory length, turning rate, traversal time and the number of true-positive incidents—the "Go-back" action is accurately triggered on encountering an obstacle—in following a forest trail from start-to-goal in the Southampton Common woodland. The experiment weather conditions and times of day are detailed in Supplementary Table S2 of the Supplementary Material.

	Trajectory length m)	Turning rate	Traversal time (mins)	True-positive "Go-back" actions
Run 1	146	0.55	23.9	7
Run 2	149	0.58	27.2	11
Run 3	150	0.54	18.7	9
Run 4	144	0.45	14.7	9
Run 5	143	0.54	18.9	10

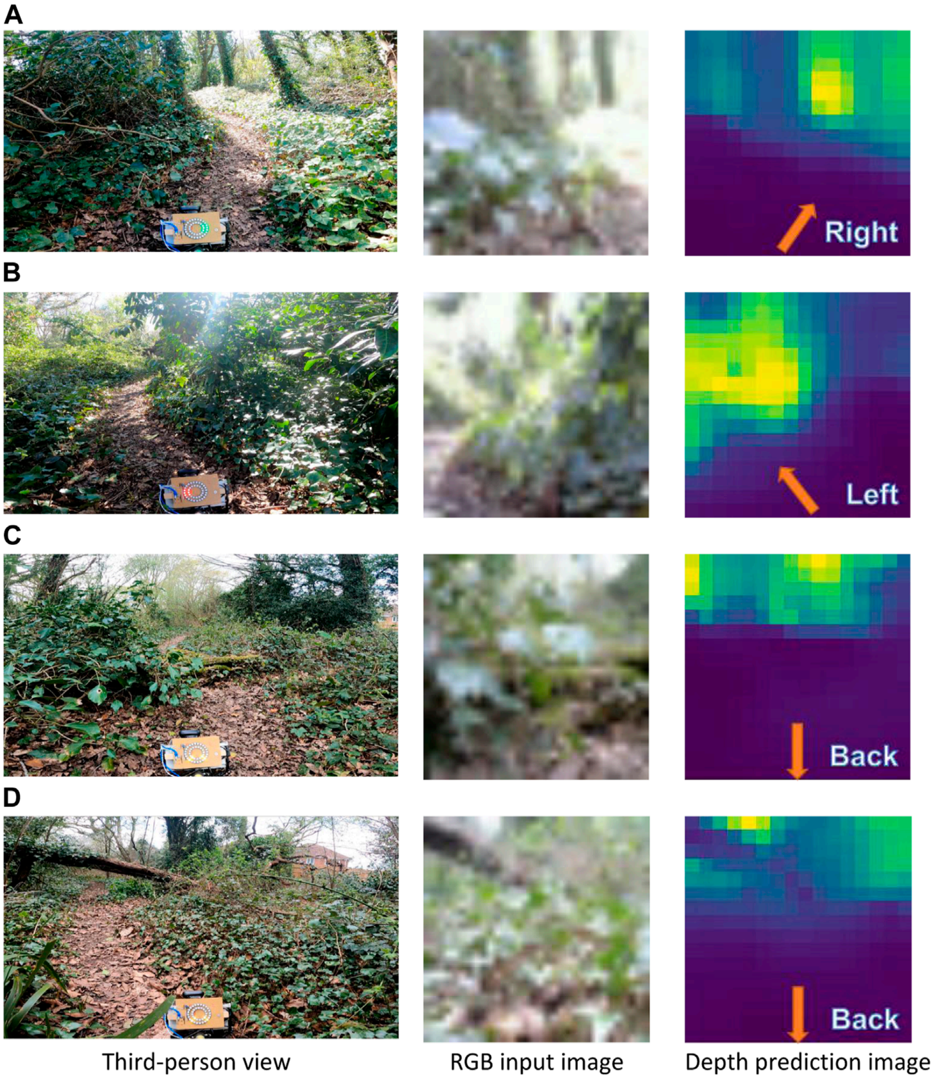

FIGURE 11
Steering directions navigated by the mobile platform on the forest trail at the Southampton Common woodland, including for clear trails and the right and left **(A, B)**, a fallen tree trunk covered in weeds and moss **(C)**, and dense vegetation **(D)**. For steering, 16 × 16 depth images, predicted from 32 × 32 RGB input images, were utilized. The corresponding 1920 × 1080 RGB images (from the GoPro camera), display a third-person view of the forest scene and the steering commands on the LED rings of the mobile platform. The steering directions are annotated on the predicted depth image. The predicted depth images displayed here have been upsampled here by a factor of two for visual clarity.

TABLE 4 The trajectory length, turning rate, traversal time and the number of true-positive incidents—the "Go-back" action is accurately triggered on encountering an obstacle—when navigating from start-to-goal off-trail in the Southampton Common woodland. Experiments in site A and site B had start waypoints at (5056.1906 N, 124.0071 W) and (5056.1814 N, 124.0148 W), respectively, and shared a common goal waypoint at (5056.1853 N, 124.0143 W). The experiment weather conditions and times of day are detailed in Supplementary Table S3 of the Supplementary Material.

	Trajectory length (m)	Turning rate	Traversal time (mins)	True-positive "Go-back" actions
Site A				
Run 1	8	0.76	2.6	0
Run 2	10	0.80	3.7	0
Run 3	13	0.62	2.5	0
Run 4	9	0.74	2.7	0
Run 5	10	0.75	3.9	2
Site B				
Run 6	24	0.73	9.3	7
Run 7	17	0.74	6.3	6
Run 8	23	0.55	5.8	6
Run 9	11	0.82	5.2	5
Run 10	19	0.72	8.2	7

Off-trail forest navigation. Experiments were performed in an unfrequented area of the woodland, spanning around 400 m^2. Obstacles on the site included forest litter, small shrubs, mounds, standing trees, fallen branches and fallen tree trunks (see examples in Figure 9B); the site was more cluttered than the forest trail environment. For our experiments, the start waypoints were located at (5056.1906 N, 124.0071 W) in site A and (5056.1814 N, 124.0148 W) in site B, in two separate and independent setups, with a common goal waypoint at (5056.1853 N, 124.0143 W).

Experiments were performed ten times in the forenoon, midday, afternoon, and near sunset, under weather conditions of cloudy, scattered clouds, mostly clear and sunny (see details in Supplementary Table S3 of the Supplementary Material). Across all experiments, the mobile platform was able to reach the goal-way point without sustaining any collisions and incurring any false-negative incidents, irrespective of the time of day and weather conditions. The platform traversed an average distance of 14 ± 6 m from start-to-goal, with a turning rate of 0.72 ± 0.08, in 5.0 ± 2.4 min (see Table 4); a higher turning rate, compared to the forest trail environment, may be due to their being more obstacles off-trail. Despite the higher density of obstacles, the mobile platform was able to avoid them with a sequence of turning actions (see examples in Figures 12A,B of the platform avoiding a slender tree and a fallen tree trunk). Moreover, as with the forest off-trail experiments, the "Go-back" action was accurately triggered to avoid potential collisions (see Figures 12C,D of a fallen tree trunk and a large fallen branch). The "Go-back" action was unnecessarily triggered only once—a false-positive incident—when the platform was facing an incline. Finally, in all experiments in Site B (start waypoint at (5056.1814 N, 124.0148 W)), the platform had to be lifted over an obstacle once; dense bushes on either end of a large fallen tree were blocking all traversable paths to reach the goal.

4 Discussion

In this study, we have implemented a low-viewpoint navigation algorithm for inexpensive small-sized mobile platforms navigating forest environments. For navigation, a depth prediction model was trained to predict depth images from RGB images of a monocular camera mounted on the mobile platform. Subsequently, a simple steering algorithm used predicted depth values to direct the platform towards open traversable areas of the forest, while avoiding obstacles. Our algorithm was extensively tested both in high-fidelity simulated forests, and real-world forests under several different weather conditions and times of day. In field experiments, using no more than a 16 × 16 depth image predicted from a 32 × 32 monocular RGB image, our mobile platform was able to successfully traverse a total of over 750 m of forest terrain comprising small shrubs, dense bushes, tall grass, fallen branches, fallen tree trunks, ditches, mounds and standing trees.

A computational bottleneck of our navigation algorithms is depth prediction, requiring around 0.8 s per RGB image with a DenseNet-121 network on a Raspberry Pi 4. The runtime for depth prediction may be improved with alternative state-of-the-art light-weight convolutional neural network architectures. For instance, the MobileNet (Howard et al., 2017), MobileNet-v2 (Sandler et al., 2018), NASNetMobile (Zoph et al., 2018), ShuffleNet (Zhang et al., 2018), ShuffleNet-v2 (Ma et al., 2018) and the Pyramidal-Depth networks (Poggi et al., 2018) may potentially improve runtime performance, consequent to their small size. However, the performance of these encoders for depth prediction at low viewpoints in forest environments remains to be investigated.

The runtime performance of our algorithms could also be enhanced with marginally more expensive embedded platforms such as the Jetson Nano, Jetson TX1 and Jetson TX2 computers, instead of the Raspberry Pi 4 (Mancini et al., 2018; An et al., 2021; Yang et al., 2021). Our results from a preliminary investigation

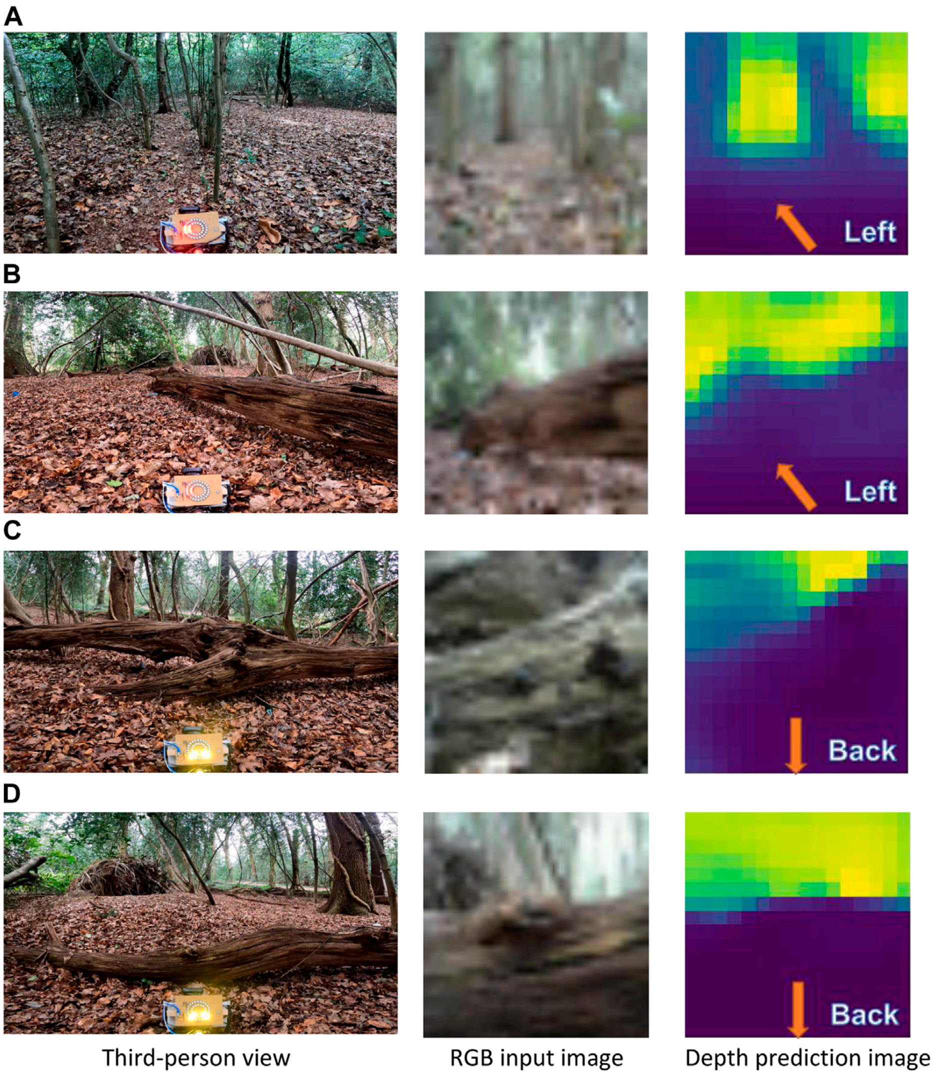

FIGURE 12
Steering directions navigated by the mobile platform off-trail at the Southampton Common woodland to avoid a number of slender trees **(A)**, and fallen tree trunks and branches **(B–D)**. The steering directions are annotated on the predicted depth image, upsampled here by a factor of two for visual clarity.

suggests a three-time increase in average runtime performance for depth prediction of forest scenes at low viewpoints with the Jetson Nano. Finally, budget permitting, the depth prediction model may bes altogether replaced with low-end RGB-D cameras, such as Microsoft Kinect, Intel Realsense, and Orbbec Astra series, thus providing high-resolution depth images, if needed, to our computationally inexpensive steering algorithm.

In challenging forest terrain, the rover may be obstructed by untraversable obstacles, such as dense vegetation, with no accessible paths to reach the destination waypoint. It may therefore be crucial to use strategies to navigate out of the local area. Integrating a robotic arm (Haddeler et al., 2022) on the rover enhances its capabilities for actively exploring such terrain. Equipped with force sensors, the arm may probe its surroundings to identify openings or traversable gaps in the terrain. The additional information may consequently inform the rover's navigation decisions, improving its ability to maneuver and find suitable paths through the obstacles.

In a few studies, aerial drones are being explored for the monitoring of forest environments (Giusti et al., 2015; Dionisio-Ortega et al., 2018; Iuzzolino et al., 2018; Sudhakar et al., 2020; Zhou et al., 2022). Arguably, rovers capable of navigating forest terrain could complement these technologies. A heterogeneous UGV-UAV team of robots could leverage the strengths of both platforms; the high vantage viewpoint afforded by the aerial drones and the close access to the ground of the rovers. Forest rovers may be adapted to provide a range of ground-based measurements such as physical samples of the soil (Zaman et al., 2022). They could pause to take reliable pollen or greenhouse gas samples close to the ground (Grau Ruiz and O'Brolchain, 2022; Gupta et al., 2022). Moreover, small-sized rovers could be operated quietly, being less intrusive to wildlife than drones (Mulero-Pázmány et al., 2017). Finally, in using drones, the end-user may face a number of hurdles in licensing the vehicles, and be limited to line-of-sight operations (Hodgkinson and Johnston, 2018).

5 Concluding remarks

In our study, a mobile platform was pushed by a human operator, following directions provided by the onboard navigation algorithm. Such a platform enabled us to focus solely on the problem of navigation in forest environments, allowing relatively rapid algorithmic prototyping. However, our navigation algorithm has the potential to successfully steer real small-sized rovers. Firstly, the monocular camera mounted inside our mobile platform enclosure is 20 cm off the ground, consistent with the low viewpoint of off-road small-sized rovers (e.g., see rovers deployed in Sebastian et al. (2019); Tang et al. (2019); Murphy et al. (2012), but with expensive sensor hardware and navigating in urban and structured off-road environments). Secondly, our navigation algorithm is robust to motion blur in the RGB images from the movement of our mobile platform, as well as high-contrast lighting, and low-lighting conditions. Finally, our navigation algorithm is tolerant to naturally occurring variations in steering angles and forward displacement step-sizes; these variations are inadvertently caused by the human operator pushing the platform, and from the platform-terrain interactions. Here we report the performance values from the field to give an orientation of how they compare to the simulations. It is important to note that such an approach can only be used to compare the performance of different algorithms if the human operator is neutral or blind to which one is running during the field experiment. The unavoidable bias then applies equally to the techniques under study. In conclusion, the challenges in designing high-endurance low-cost rover hardware capable of self-moving on challenging forest terrain are immense, and are being tackled by us in a separate study (Tarapore et al., 2020); in our future work, we will investigate the performance of our navigation algorithm on real rovers.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

CNi, DT, and K-PZ designed the study. CNe developed the forest simulator. CNi performed all the simulation and field experiments. CNi and DT wrote the manuscript. All authors contributed to the article and approved the submitted version.

Acknowledgments

The authors acknowledge the use of the IRIDIS High-Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The supplementary material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2023.1086798/full#supplementary-material

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). *TensorFlow: Large-Scale machine learning on heterogeneous distributed systems.* doi:10.48550/ARXIV.1603.04467

Ai, B., Gao, W., Vinay, , and Hsu, D. (2022). "Deep visual navigation under partial observability," in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23-27 May 2022 (IEEE), 9439–9446. doi:10.1109/ICRA46639.2022.9811598

Alhashim, I., and Wonka, P. (2018). *High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941.* doi:10.48550/ARXIV.1812.11941

An, S., Zhou, F., Yang, M., Zhu, H., Fu, C., and Tsintotas, K. A. (2021). "Real-time monocular human depth estimation and segmentation on embedded systems," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 Sept.-1 Oct. 2021 (IEEE), 55–62. doi:10.1109/IROS51168.2021.9636518

Bagnell, J. A., Bradley, D., Silver, D., Sofman, B., and Stentz, A. (2010). Learning for autonomous navigation. *IEEE Robotics Automation Mag.* 17, 74–84. doi:10.1109/MRA.2010.936946

Baril, D., Deschênes, S.-P., Gamache, O., Vaidis, M., LaRocque, D., Laconte, J., et al. (2022). Kilometer-scale autonomous navigation in subarctic forests: Challenges and lessons learned. *Field Robot.* 2, 1628–1660. doi:10.55417/fr.2022050

Batey, T. (2009). Soil compaction and soil management – A review. *Soil Use Manag.* 25, 335–345. doi:10.1111/j.1475-2743.2009.00236.x

Bellone, M., Reina, G., Caltagirone, L., and Wahde, M. (2017). Learning traversability from point clouds in challenging scenarios. *IEEE Trans. Intelligent Transp. Syst.* 19, 296–305. doi:10.1109/TITS.2017.2769218

Bjelonic, M., Kottege, N., Homberger, T., Borges, P., Beckerle, P., and Chli, M. (2018). Weaver: Hexapod robot for autonomous navigation on unstructured terrain. *J. Field Robotics* 35, 1063–1079. doi:10.1002/rob.21795

Borges, P., Peynot, T., Liang, S., Arain, B., Wildie, M., Minareci, M., et al. (2022). A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges. *Field Robot.* 2, 1567–1627. doi:10.55417/fr.2022049

Braun, T., Schaefer, H., and Berns, K. (2009). "Topological large-scale off-road navigation and exploration RAVON at the European land robot trial 2008," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 10-15 Oct. 2009 (IEEE), 4387–4392. doi:10.1109/IROS.2009.5353985

Carvalho, A. E., Ferreira, J. F., and Portugal, D. (2022). "3D traversability analysis in forest environments based on mechanical effort," in International Conference on Intelligent Autonomous Systems (IAS-17), Cham: Springer Nature Switzerland, 457–468. doi:10.1007/978-3-031-22216-0_17

Chen, D., Zhuang, M., Zhong, X., Wu, W., and Liu, Q. (2022). Rspmp: Real-time semantic perception and motion planning for autonomous navigation of unmanned ground vehicle in off-road environments. *Appl. Intell.*, 1–17. doi:10.1007/s10489-022-03283-z

Corke, P., Paul, R., Churchill, W., and Newman, P. (2013). "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3-7 Nov. 2013 (IEEE), 2085–2092. doi:10.1109/IROS.2013.6696648

da Silva, D. Q., dos Santos, F. N., Sousa, A. J., Filipe, V., and Boaventura-Cunha, J. (2021). Unimodal and multimodal perception for forest management: Review and dataset. *Computation* 9, 127. doi:10.3390/computation9120127

Desa, U. (2021). *The global forest goals report 2021*. United Nations Publication. doi:10.18356/9789214030515

Dionisio-Ortega, S., Rojas-Perez, L. O., Martinez-Carranza, J., and Cruz-Vega, I. (2018). "A deep learning approach towards autonomous flight in forest environments," in 2018 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 21-23 Feb. 2018 (IEEE), 139–144. doi:10.1109/CONIELECOMP.2018.8327189

Fnadi, M., Du, W., da Silva, R. G., Plumet, F., and Benamar, F. (2020). "Local obstacle-skirting path planning for a fast Bi-steerable rover using bézier curves," in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May-31 Aug. 2020 (IEEE), 10242–10248. doi:10.1109/ICRA40945.2020.9197563

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.* 32, 1231–1237. doi:10.1177/0278364913491297

Gerdes, L., Azkarate, M., Sánchez-Ibáñez, J. R., Joudrier, L., and Perez-del Pulgar, C. J. (2020). Efficient autonomous navigation for planetary rovers with limited resources. *J. Field Robotics* 37, rob.21981–1170. doi:10.1002/rob.21981

Giusti, A., Guzzi, J., Cireşan, D. C., He, F.-L., Rodríguez, J. P., Fontana, F., et al. (2015). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics Automation Lett.* 1, 661–667. doi:10.1109/LRA.2015.2509024

Gonzalez, R., Rodriguez, F., Guzman, J. L., Pradalier, C., and Siegwart, R. (2013). Control of off-road mobile robots using visual odometry and slip compensation. *Adv. Robot.* 27, 893–906. doi:10.1080/01691864.2013.791742

González, R., Rodríguez, F., Sánchez-Hermosilla, J., and Donaire, J. G. (2009). Navigation techniques for mobile robots in greenhouses. *Appl. Eng. Agric.* 25, 153–165. doi:10.13031/2013.26324

Grau Ruiz, M. A., and O'Brolchain, F. (2022). Environmental robotics for a sustainable future in circular economies. *Nat. Mach. Intell.* 4, 3–4. doi:10.1038/s42256-021-00436-6

Gupta, A., van der Schoor, M. J., Bräutigam, J., Justo, V. B., Umland, T. F., and Göhlich, D. (2022). Autonomous service robots for urban waste management-multiagent route planning and cooperative operation. *IEEE Robotics Automation Lett.* 7, 8972–8979. doi:10.1109/LRA.2022.3188900

Haddeler, G., Chan, J., You, Y., Verma, S., Adiwahono, A. H., and Meng Chew, C. (2020). "Explore bravely: Wheeled-legged robots traverse in unknown rough environment," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 Oct.-24 Jan. 2021 (IEEE), 7521–7526. doi:10.1109/IROS45743.2020.9341610

Haddeler, G., Chuah, M. Y. M., You, Y., Chan, J., Adiwahono, A. H., Yau, W. Y., et al. (2022). Traversability analysis with vision and terrain probing for safe legged robot navigation. *Front. Robotics AI* 9, 887910. doi:10.3389/frobt.2022.887910

Ho, K., Peynot, T., and Sukkarieh, S. (2013a). "A near-to-far non-parametric learning approach for estimating traversability in deformable terrain," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3-7 Nov. 2013 (IEEE), 2827–2833. doi:10.1109/IROS.2013.6696756

Ho, K., Peynot, T., and Sukkarieh, S. (2013b). "Traversability estimation for a planetary rover via experimental kernel learning in a Gaussian process framework," in 2013 IEEE International Conference on Robotics and Automation (ICRA), 3475–3482. doi:10.1109/ICRA.2013.6631063

Hodgkinson, D., and Johnston, R. (2018). *Aviation law and drones: Unmanned aircraft and the future of aviation*. Routledge. doi:10.4324/9781351332323

Hoeller, F., Röhling, T., and Schulz, D. (2013). "Collective motion pattern scaling for improved open-loop off-road navigation," in 2013 13th International Conference on Autonomous Robot Systems (ICARS), Lisbon, Portugal, 24-24 April 2013 (IEEE), 1–6. doi:10.1109/Robotica.2013.6623535

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. doi:10.48550/ARXIV.1704.04861

Howard, T. M., Green, C. J., Kelly, A., and Ferguson, D. (2008). State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *J. Field Robotics* 25, 325–345. doi:10.1002/rob.20244

Howard, T. M., Green, C. J., and Kelly, A. (2010). Receding horizon model-predictive control for mobile robot navigation of intricate paths. *Field Serv. Robotics Results 7th Int. Conf. Springer Berlin Heidelberg* 69, 69–78. doi:10.1007/978-3-642-13408-1_7

Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21-26 July 2017 (IEEE), 2261–2269. doi:10.1109/CVPR.2017.243

Huang, W., Grudic, G., and Matthies, L. (2009b). Editorial. *J. Field Robotics* 26, 115–116. doi:10.1002/rob.20280

Huang, W., Grudic, G., and Matthies, L. (2009a). Editorial: Editorial. *J. Field Robotics* 26, 1–2. doi:10.1002/rob.20275

Iuzzolino, M. L., Walker, M. E., and Szafir, D. (2018). "Virtual-to-real-world transfer learning for robots on wilderness trails," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1-5 Oct. 2018 (IEEE), 576–582. doi:10.1109/IROS.2018.8593883

Jackel, L. D., Krotkov, E., Perschbacher, M., Pippine, J., and Sullivan, C. (2006). The DARPA LAGR program: Goals, challenges, methodology, and phase I results. *J. Field robotics* 23, 945–973. doi:10.1002/rob.20161

Karpyshev, P., Ilin, V., Kalinov, I., Petrovsky, A., and Tsetserukou, D. (2021). "Autonomous mobile robot for apple plant disease detection based on cnn and multi-spectral vision system," in 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Fukushima, Japan, 11-14 Jan. 2021 (IEEE), 157–162. doi:10.1109/IEEECONF49454.2021.9382649

Kingma, D. P., and Ba, J. (2014). *Adam: A method for stochastic optimization*. doi:10.48550/ARXIV.1412.6980

Kragh, M., and Underwood, J. (2020). Multimodal obstacle detection in unstructured environments with conditional random fields. *J. Field Robotics* 37, 53–72. doi:10.1002/rob.21866

Krebs, A., Pradalier, C., and Siegwart, R. (2010). Adaptive rover behavior based on online empirical evaluation: Rover–terrain interaction and near-to-far learning. *J. Field Robotics* 27, 158–180. doi:10.1002/rob.20332

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi:10.1145/3065386

Krotkov, E., Fish, S., Jackel, L., McBride, B., Perschbacher, M., and Pippine, J. (2007). The DARPA perceptor evaluation experiments. *Aut. Robots* 22, 19–35. doi:10.1007/s10514-006-9000-0

Lee, H., and Chung, W. (2021). "A self-training approach-based traversability analysis for mobile robots in urban environments," in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May-5 June 2021 (IEEE), 3389–3394. doi:10.1109/ICRA48506.2021.9561394

Lee, J., Park, G., Cho, I., Kang, K., Pyo, D., Cho, S., et al. (2022). ODS-Bot: Mobile robot navigation for outdoor delivery services. *IEEE Access* 10, 107250–107258. doi:10.1109/ACCESS.2022.3212768

Liu, F., Li, X., Yuan, S., and Lan, W. (2020a). Slip-aware motion estimation for off-road mobile robots via multi-innovation unscented kalman filter. *IEEE Access* 8, 43482–43496. doi:10.1109/ACCESS.2020.2977889

Liu, L., Dugas, D., Cesari, G., Siegwart, R., and Dubé, R. (2020b). "Robot navigation in crowded environments using deep reinforcement learning," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 Oct.-24 Jan. 2021 (IEEE), 5671–5677. doi:10.1109/IROS45743.2020.9341540

Louis, M. S., Azad, Z., Delshadtehrani, L., Gupta, S., Warden, P., Reddi, V. J., et al. (2019). Towards deep learning using TensorFlow lite on RISC-V. *Third Workshop Comput. Archit. Res. RISC-V (CARRV)* 1, 6. doi:10.13140/RG.2.2.30400.89606

Lourenço, D., De Castro Cardoso Ferreira, J., and Portugal, D. (2020). "3D local planning for a forestry UGV based on terrain gradient and mechanical effort," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Workshop on Perception, Planning and Mobility in Forestry Robotics (WPPMFR 2020), Las Vegas, NV, USA, Oct 29 (2020) (IEEE).

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. (2018). "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in Proceedings of the European conference on computer vision (ECCV), 116–131. doi:10.1007/978-3-030-01264-9_8

Mancini, M., Costante, G., Valigi, P., and Ciarfuglia, T. A. (2018). J-mod2: Joint monocular obstacle detection and depth estimation. *IEEE Robotics Automation Lett.* 3, 1490–1497. doi:10.1109/LRA.2018.2800083

Mayuku, O., Surgenor, B. W., and Marshall, J. A. (2021). "A self-supervised near-to-far approach for terrain-adaptive off-road autonomous driving," in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May-5 June 2021 (IEEE), 14054–14060. doi:10.1109/ICRA48506.2021.9562029

Milella, A., Reina, G., and Underwood, J. (2015). A self-learning framework for statistical ground classification using radar and monocular vision. *J. Field Robotics* 32, 20–41. doi:10.1002/rob.21512

Mulero-Pázmány, M., Jenni-Eiermann, S., Strebel, N., Sattler, T., Negro, J. J., and Tablado, Z. (2017). Unmanned aircraft systems as a new source of disturbance for wildlife: A systematic review. *PloS one* 12, e0178448. doi:10.1371/journal.pone.0178448

Murphy, L., Martin, S., and Corke, P. (2012). "Creating and using probabilistic costmaps from vehicle experience," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 7-12 Oct. 2012 (IEEE), 4689–4694. doi:10.1109/IROS.2012.6386118

Nathan Silberman, P. K., Hoiem, D., and Fergus, R. (2012). "Indoor segmentation and support inference from RGB-D images," in European Conference on Computer Vision (ECCV), 746–760. doi:10.1007/978-3-642-33715-4_54

Newlands, C., and Zauner, K.-P. (2022). *Procedural generation and rendering of realistic, navigable forest environments: An open-source tool.* doi:10.48550/ARXIV.2208.01471

Niu, C., Tarapore, D., and Zauner, K.-P. (2020). "Low-viewpoint forest depth dataset for sparse rover swarms," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, 24 Oct.-24 Jan. 2021 (IEEE), 8035–8040. doi:10.1109/IROS45743.2020.9341435

Ostafew, C. J., Schoellig, A. P., and Barfoot, T. D. (2016). Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking. *Int. J. Robotics Res.* 35, 1547–1563. doi:10.1177/0278364916645661

Palazzo, S., Guastella, D. C., Cantelli, L., Spadaro, P., Rundo, F., Muscato, G., et al. (2020). "Domain adaptation for outdoor robot traversability estimation from RGB data with safety-preserving loss," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 Oct.-24 Jan. 2021 (IEEE), 10014–10021. doi:10.1109/IROS45743.2020.9341044

Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Eng. Appl. Artif. Intell.* 26, 1373–1385. doi:10.1016/j.engappai.2013.01.006

Peynot, T., Lui, S.-T., McAllister, R., Fitch, R., and Sukkarieh, S. (2014). Learned stochastic mobility prediction for planning with control uncertainty on unstructured terrain. *J. Field Robotics* 31, 969–995. doi:10.1002/rob.21536

Poggi, M., Aleotti, F., Tosi, F., and Mattoccia, S. (2018). "Towards real-time unsupervised monocular depth estimation on CPU," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1-5 Oct. 2018 (IEEE), 5848–5854. doi:10.1109/IROS.2018.8593814

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "MobileNetV2: Inverted residuals and linear bottlenecks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18-23 June 2018 (IEEE), 4510–4520. doi:10.1109/CVPR.2018.00474

Santamaria-Navarro, À., Teniente, E. H., Morta, M., and Andrade-Cetto, J. (2015). Terrain classification in complex three-dimensional outdoor environments. *J. Field Robotics* 32, 42–60. doi:10.1002/rob.21521

Santana, P., and Correia, L. (2011). Swarm cognition on off-road autonomous robots. *Swarm Intell.* 5, 45–72. doi:10.1007/s11721-010-0051-7

Schilling, F., Chen, X., Folkesson, J., and Jensfelt, P. (2017). "Geometric and visual terrain classification for autonomous mobile navigation," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24-28 Sept. 2017 (IEEE), 2678–2684. doi:10.1109/IROS.2017.8206092

Sebastian, B., Ren, H., and Ben-Tzvi, P. (2019). "Neural network based heterogeneous sensor fusion for robot motion planning," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3-8 Nov. 2019 (IEEE), 2899–2904. doi:10.1109/IROS40897.2019.8967689

Silver, D., Bagnell, J. A., and Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *Int. J. Robotics Res.* 29, 1565–1592. doi:10.1177/0278364910369715

Sudhakar, S., Vijayakumar, V., Kumar, C. S., Priya, V., Ravi, L., and Subramaniyaswamy, V. (2020). Unmanned aerial vehicle (uav) based forest fire detection and monitoring for reducing false alarms in forest-fires. *Comput. Commun.* 149, 1–16. doi:10.1016/j.comcom.2019.10.007

Tang, Y., Cai, J., Chen, M., Yan, X., and Xie, Y. (2019). "An autonomous exploration algorithm using environment-robot interacted traversability analysis," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3-8 Nov. 2019 (IEEE), 4885–4890. doi:10.1109/IROS40897.2019.8967940

Tarapore, D., Groß, R., and Zauner, K.-P. (2020). Sparse robot swarms: Moving swarms to real-world applications. *Front. Robotics AI* 7, 83. doi:10.3389/frobt.2020.00083

Ugenti, A., Vulpi, F., Domínguez, R., Cordes, F., Milella, A., and Reina, G. (2022). On the role of feature and signal selection for terrain learning in planetary exploration robots. *J. Field Robotics* 39, 355–370. doi:10.1002/rob.22054

Wellhausen, L., and Hutter, M. (2021). "Rough terrain navigation for legged robots using reachability planning and template learning," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 Sept.-1 Oct. 2021 (IEEE), 6914–6921. doi:10.1109/IROS51168.2021.9636358

Wolf, P., Ropertz, T., Oswald, M., and Berns, K. (2018). "Local behavior-based navigation in rough off-road scenarios based on vehicle kinematics," in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21-25 May 2018 (IEEE), 719–724. doi:10.1109/ICRA.2018.8460631

Yang, G.-Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of Science Robotics. *Sci. Robotics* 3, eaar7650. doi:10.1126/scirobotics.aar7650

Yang, X., Chen, J., Dang, Y., Luo, H., Tang, Y., Liao, C., et al. (2021). Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network. *IEEE Trans. Intelligent Transp. Syst.* 22, 156–167. doi:10.1109/TITS.2019.2955598

Zaman, A., Ashraf, F., Khan, H., Noshin, F., Samir, O., Rayhan, A. M., et al. (2022). *Mbldp-r: A multiple biomolecules based rapid life detection protocol embedded in a rover scientific subsystem for soil sample analysis*. doi:10.21203/rs.3.rs-1263964/v1

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18-23 June 2018 (IEEE), 6848–6856. doi:10.1109/CVPR.2018.00716

Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., et al. (2022). Swarm of micro flying robots in the wild. *Sci. Robotics* 7, eabm5954. doi:10.1126/scirobotics.abm5954

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). "Learning transferable architectures for scalable image recognition," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18-23 June 2018 (IEEE), 8697–8710. doi:10.1109/CVPR.2018.00907