IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Building Energy-Cost Maps from Aerial Images and Ground Robot Measurements with Semi-supervised Deep Learning

Minghan Wei[1] and Volkan Isler[2]

***Abstract*— Planning energy-efficient paths is an important capability in many robotics applications. Obtaining an energy-cost map for a given environment enables planning such paths between any given pair of locations within the environment. However, efficiently building an energy map is challenging, especially for large environments. Some of the prior work uses physics-based laws (friction and gravity force) to model energy costs across environments. These methods work well for uniform surfaces, but they do not generalize well to uneven terrains. In this paper, we present a method to address this mapping problem in a data-driven fashion for the cases where an aerial image of the environment can be obtained. To efficiently build an energy-cost map, we train a neural network that learns to predict the complete energy maps by combining aerial images and sparse ground robot energy-consumption measurements. Field experiments are performed to validate our results. We show that our method can efficiently build an energy-cost map accurately even across different types of ground robots.**

I. INTRODUCTION

Most autonomous ground robots are powered by batteries with limited capacity. When they perform field tasks, the energy-cost map of the field is helpful since, from the map, we can calculate an energy-efficient path between any two locations. The total operation time of the robot can then be improved. However, it is still challenging to build an energy-cost map, especially for large environments. To overcome this issue, some work in the literature estimates the friction force to model the energy cost of robot motion [18]. Unfortunately, if the environment is non-uniform, measuring the friction coefficient for every location in the given environment is tedious and time-consuming. Further, when the ground is not a flat surface, the jerky motion of the robot consumes more energy, which cannot be modeled simply by either friction or gravity forces.

In this paper, we address the problem of efficiently building an energy-cost map for ground robot path planning. We build on our previous work [20] [21] where we used a data-driven approach and showed that an energy-cost map can be efficiently learned by combining aerial images with ground robot energy-consumption measurements. Rather than traversing the entire environment to collect energy data, we made use of the correlations between the energy cost and the terrain appearance. We showed that energy costs of nearby locations *within the same terrain class* are close [21]. This allowed us to build the entire energy map from a sparse set of ground measurements.

[1, 2]The authors are with the Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN, US, 55108 weixx526, isler@umn.edu.

Our previous method relied on segmenting the terrain using an independently trained network. As a result, if we are given a new environment with a different set of terrain classes, the previous training results of these methods are no longer valid. The need to train a segmentation network from scratch was the main disadvantage of our previous method. One way to overcome this limitation is to train a network that can segment all the possible terrain classes, since the aerial images are easily accessible. However, energy consumption depends also on robot characteristics. Using the same ground robot to collect energy-consumption measurements on all possible terrain classes is not an easy task. Another simple solution is to collect energy measurements with one ground robot as ground truth, and then train with RGB images to predict energy-cost maps. But if we use a different robot, we need to use the new robot to take measurements on terrains and use the new measurements to retrain networks, which is time-consuming and not feasible.

In this paper, we address this limitation of relying on a pre-segmented image by estimating the energy map directly from the image (rather than the segmentation) using a semi-supervised approach. Fig. 1 gives an overview of our method. It is based on training a neural network which takes as input an aerial image along with energy measurements at locations corresponding to a small fraction of pixels $(1 - 2\%)$. The present work improves our previous work in two ways: 1) It outputs the energy-cost map directly from ground appearance and a sparse set of measurements without terrain segmentation. This way, it generalizes to environments with unknown terrain classes. 2) The network can also be fine-tuned for new ground robots. We perform field experiments to validate our results. Using the energy-cost map reconstructed by our method, our experiments show that our method finds more energy-efficient paths compared to shortest-path planner.

II. RELATED WORK

Improving ground robot navigation algorithms is gaining importance in robotics research as robots become common in applications such as precision agriculture, environmental monitoring, search, and mapping. Considering that many ground robots are powered by batteries with limited budgets, how to navigate in an energy-efficient manner is also widely studied.

Given an environment in which the robot navigates, if the energy-cost map of this environment is available, energy-optimal path between any pair of locations can be calculated using Dijkstra's algorithm assuming worlds are well-represented by a grid. RRT algorithm [9] or its variants [8],

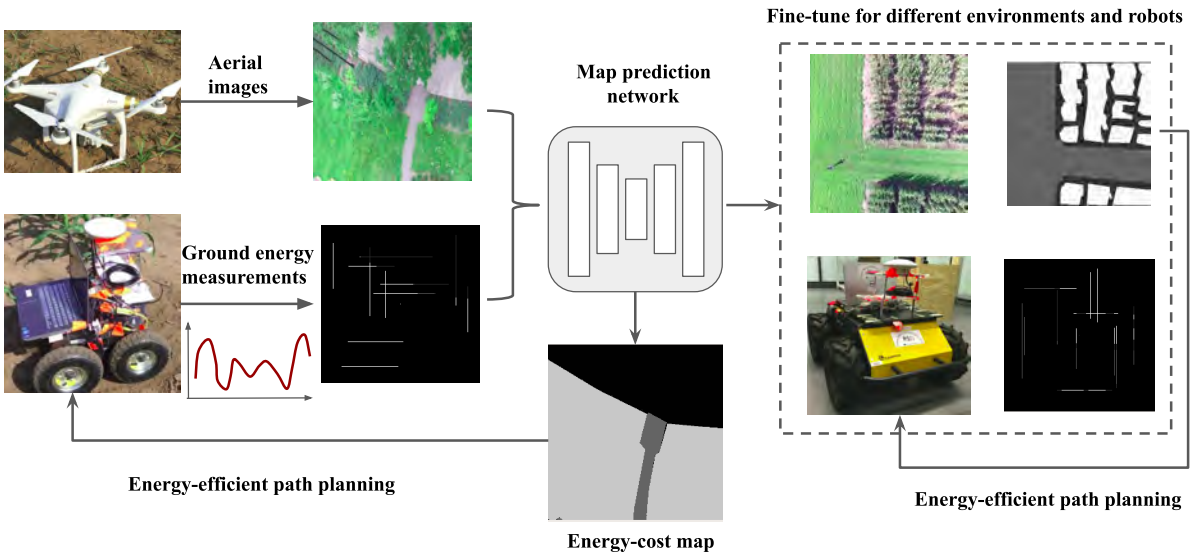


Fig. 1. An overview of the method in this paper. The network learns the energy-cost map from aerial images and sparse ground robot energy-consumption measurements. The trained network can estimate the map for planning energy-efficient paths. The network can be fine-tuned for planning in different environments with different ground robots.

[1] can also be applied to improve the planning time. D^* algorithm is used for the case where the map is not fully observed and can change during navigation [7]. We refer the readers to surveys [4], [11], [5] for additional path planning methods.

However, building an energy-cost map is not an easy task. Some prior work uses physics-based laws (friction and gravity force) to model the energy consumption of the robot [18], [17], [10]. Since obtaining environment properties such as friction coefficient can be difficult, we explored data-driven approaches [20], [21], to build the map by combining air and ground measurements.

Since acceleration and turning usually cost more energy compared to constant-velocity movement, the velocity profile also affects navigation efficiency. Mei et al. related the power consumption with the motor speed using polynomials and demonstrated significant energy savings in simulation [12]. Tokekar et al. studied a more complex energy-cost model by considering various components such as the internal and load friction, power to the output shaft [19]. They presented an algorithm to find optimal velocity profiles.

Visual inputs can provide general information about the environment, thus they are also used for ground robot navigation. A feasible path can be found by learning the ground traversability from images [16], [2]. Delmerico et al. proposed an algorithm for ground and aerial vehicles moving simultaneously to explore an environment [3]. In this paper, we present a semi-supervised learning method to estimate energy consumption directly from aerial maps.

III. Preliminaries and Problem Formulation

We consider the problem of navigating a ground robot in a given environment. The robot is able to measure its energy consumption during the movement. The energy cost of a given trajectory is calculated by:

$$E = \int_{x_0}^{x_t} e(x, s)dx \tag{1}$$

where e is the unit-distance energy consumption, as defined in [21], at location x. The value of e depends on the location (environment properties), as well as the robot state s (velocity, orientation, etc). Previous work in the literature studied optimizing energy cost considering the robot states [19]. In our work, we focus on environment effects and assume the robot moves at a constant speed. In this way, we can simply denote $e(x, s)$ as $e(x)$. $e(x)$ can have multiple values for different moving directions due to the slope at x. To avoid feeding multiple $e(x)$ values for all the locations into the network, in this paper, we focus on environments of zero inclination angle. For areas with slopes, our method can be extended as follows: we can add an extra energy cost of $E' = mgl \sin\theta$ due to the gravity where mg is the weight, l is the distance, and θ is the inclination angle. There are several ways to extract the slope (inclination) information, e.g., 3D reconstruction with the aerial images.

The energy-cost map consists of unit-distance energy consumption of every location in the environment. This map is unknown in the beginning. Our goal in this paper is to build such a map efficiently so that given any pair of locations in the environment, we can plan an energy-efficient path for the ground robot.

We rely on the assumption that the unit-distance energy cost on the same terrain class is a constant plus noise for the ground robot. We present an overview of our method in Sec. IV. Next, we explain our data collection procedure, followed by additional network and training details. We also present practical applications of our method in Sec. V.

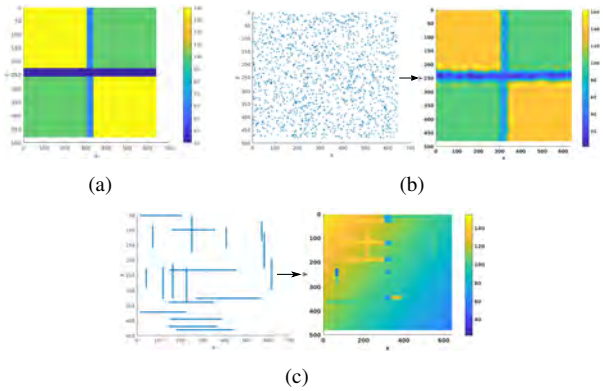


Fig. 2. Reconstructing the energy-cost map using Gaussian process regression by measuring 1% locations in the map. (a) The ground truth of the map. (b) The measurement locations are uniformly distributed in the map (left). The estimated map is close to the ground truth (right). (c) The measurement locations are a few random transects in the map (left). GP cannot estimate the map with these transects (right).

IV. Method

Our goal is to learn the energy-cost map from aerial images and sparse ground-based energy measurements in a semi-supervised fashion. In this section, we present our method.

A. Collecting Ground Measurements

In this subsection, we present the details of our ground data-collection procedure. These measurements are used to predict energy-cost maps.

Supposing the ground robot moves at a constant speed, we measure the unit-distance energy consumption, as defined in Sec. III, in the following way. The robot's total energy-consumption E is calculated by integrating the product of its motor voltage and current over time. Dividing E by the moving distance, we obtain the unit-distance energy consumption. For simplicity, in the rest of the paper, the term *ground measurement* refers to measuring the unit-distance energy costs.

One simple way to select ground measurement locations is to make them evenly distributed. To reconstruct the entire map from sparse evenly-distributed measurements, Gaussian process regression has been widely used in the literature. For example, in [13], [14], it is used for reconstructing solar maps. Fig. 2 demonstrates an example of using this method to build the map from sparse measurements. Fig. 2(a) is the ground truth of the map. Fig. 2(b) shows that with 1% uniformly distributed measurements (left), the reconstructed map is close to the ground truth.

However, since a ground robot can only continuously move, if the selected measurement locations are evenly distributed in the environment, it may have to cover the whole area to visit them, which is not a feasible solution. For ground robots, it is easy for them to take measurements along a trajectory. Therefore, we use transects in images as measurement locations. The transects (trajectories) are selected in the following way. We first generate the number of transects randomly in the interval of $[6, 16]$. Then for each transect, we generate a starting location, orientation and length. The length is randomly assigned between 60 and 120 pixel distance. All the transects are either horizontal or vertical. In our work, the aerial images are rescaled to 256×256. Thus the total number of measured pixels occupies on average 1.5% of the whole area. Note that we use transects of one-pixel width, though robot footprints can occupy a width of several pixels. The transects are corresponding to robot center positions when they take measurements.

The environment can contain untraversable terrains. The robot cannot take actual measurements in such areas. To incorporate traversability into our method, the aerial images can be preprocessed to detect untraversable regions using methods such as [16], [2]. Once untraversable areas are detected, we can assign a high energy cost to them.

We also note that GP regression does not work well for the case of transects as shown in Fig. 2(c). Therefore, we combine the measurements with the ground appearance using a semi-supervised learning approach which we describe next.

B. Overview

An overview of our training and evaluation process is shown in Fig. 3. Our data consists of aerial images with a small fraction $(1-2\%)$ of energy-cost measurements in the images from transects.

The training data is collected at a cornfield located in Saint Paul, Minnesota. The aerial images are collected with a DJI Phantom 3 Professional UAV by flying and covering the field at the height of 15 meters (m) from the ground. The solid blue rectangle on the map in Fig. 4(a) marks the area where we collect data. The ground robot we use is from 'Rover Robotics', as shown in Fig. 4(b).

Our procedure to separate training and evaluation (test) datasets is as follows: In the training set, we leave out 50% of the images together with the transects completely for validation during the training. To test the network performance, we further collected two more datasets. One is from an area similar to the one for training (Sec. IV-E). The other is from an area containing scenes not in the training (Sec. IV-F). We test the network performance on these two sets without including them in the training set. After this procedure, our training dataset contains 79 images with measurements, whereas our evaluation dataset contains 439 images in total.

The network input consists of RGB images and the corresponding energy measurements. To train the network, we interpolate the energy-cost maps of the images in the training set, so that energy-cost values at all the pixels in the images are available for training. The interpolated maps are used as ground truth for training. We interpolate maps based on the measurements on transects and terrain segmentation as follows: we calculate the unit distance energy-cost for each terrain class, by dividing the total energy cost E by the total length L on this terrain class. We use this value for all pixels belonging to this class. For terrain segmentation, we manually label the terrain classes in each training image. For untraversable terrain classes, we set high energy costs.

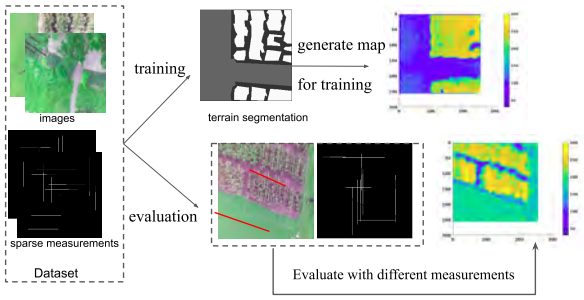
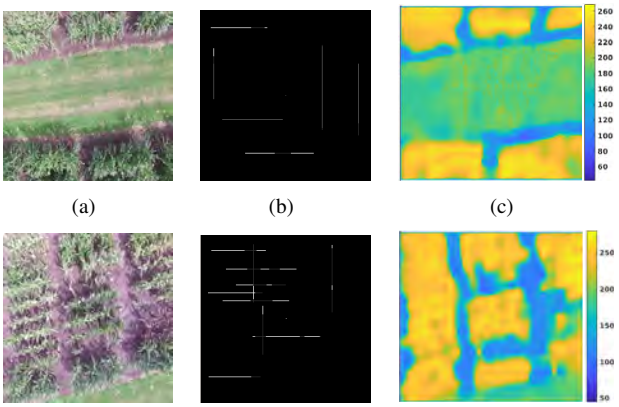


Fig. 3. The overview of our method. We split the data into training and evaluation (test) sets. For the training set, we interpolate the entire map based on the measurements and terrain segmentation. We evaluate the network performance in the test set to see if the network outputs are consistent with the ground measurements

(a) (b)

Fig. 4. We collected data from a cornfield with a ground robot. (a) We trained the network with the data from the area marked by the solid blue rectangle. The dashed red area is used for testing. (b) The ground robot we use in the cornfield.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Example map building results on the test set. Our method reconstructs them well from the images and sparse measurements. We cannot take measurements in untraversable areas. For training, we just assign a high energy-cost value to these areas so that they can be avoided during navigation. $(a)(d)$. The original images. $(b)(e)$ The sparse sample measurements. $(c)(f)$. The energy-cost map from the network output.

Note that labeling the entire map is only required for obtaining ground truth to train the network. During the test stage, we just need to take measurements at locations selected by the method in Sec. IV-A as network inputs. To evaluate the network performance (Sec. IV-D), we use ground robots to take actual cost measurements at locations not selected by the training transects. We compare these measurements with the network outputs.

C. Network Architecture

An encoder-decoder network structure, such as U-Net [15], has been widely used in detection and segmentation tasks. In this paper, we use a similar architecture with modifications to the input and output layers.

The input to our network is of size $256 \times 256 \times 4$, where 256×256 corresponds to the aerial image size, and the four channels are RGB and the ground energy-cost measurements. The images are normalized before being input to the network. For the energy-cost channel, the pixels without measurements are set to zero. There are two network structures that can be used to process multi-channel inputs [6]: we can either fuse the four channels together using a convolutional layer in the beginning of the network (early-fusion) , or feed the RGB channel and the measurement channel to separate layers and fuse them later (late-fusion). In this paper we use the early-fusion approach as it yields a simpler design.

The output size is 256×256, where each value is the prediction of the energy cost at the corresponding pixel position. We use mean square error as the loss function for regressing the energy predictions. The Adam optimizer with a step size of 10^{-4} is used for training the network.

D. Training Details and Results

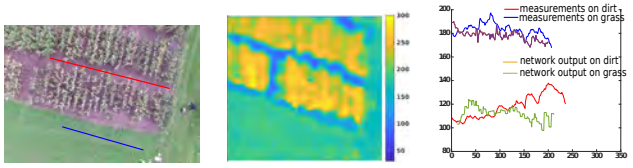
We extract in total 241 images covering the field. We label 79 of them and use half of them for training the network and the other half for validation. It takes the robot $117J/m$ to move on the dirt roads, and $185J/m$ on grasses in this area on average. The corn plants are non-traversable. We assign a high energy-cost value to them $(250J/m)$. We use these values to interpolate the energy-cost maps, as explained in Sec. IV-B.

The network is trained on a Dell Precision 7530 machine with a Nvidia Quadro P3200 GPU. When training from scratch, it takes around one hour for the training loss per pixel to drop within 2% of the ground truth value.

Fig. 5 demonstrates two map building results on the test set. We evaluate the network performance by collecting energy-cost data at transects that are not selected as network inputs. We compare the data with the network outputs to see if they are consistent. Fig. 6 shows an example of evaluating the network performance. The collected unit-distance energy cost measurements in Fig. 6(c) fluctuate around the values in the map at corresponding positions, which validates our method.

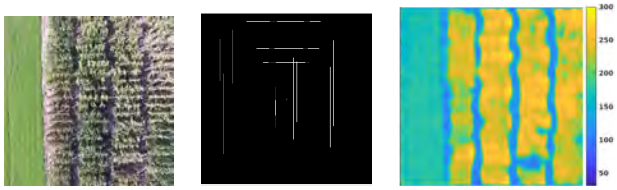
E. Generalization to Similar Environments and Ground Robots

In this subsection, we show that the pre-trained network can be directly applied to an area similar to the training



(a) (b) (c)

Fig. 6. An example of evaluating the network performance with measurements at different transects.



(a) (b) (c)

Fig. 7. An example of building the energy-cost map for a different part of the cornfield. (a) One of the aerial images of this field. (b) The sparse measurements using the ground robot. (c) The estimated energy-cost map. It reflects the energy-cost differences among terrain classes.

environment with a ground robot of similar energy costs. The training and testing environments are selected from two parts of a cornfield, which is marked in Fig. 4(a). We directly apply the network trained in Sec. IV-D for testing.

Fig. 7 shows a map building example for this part of the cornfield. From Fig. 7(c) we can see that the estimated map is accurate.

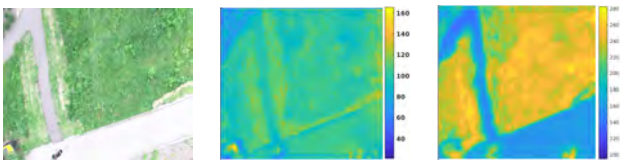
F. Generalization to New Environments and Ground Robots

In this subsection, we test if the pre-trained results can be used to generate energy-cost maps for completely new environments and ground robots, without training new networks from scratch.

To show that the training result is useful for an environment with terrain classes not contained in the training set, we apply the pre-trained network to the dataset from a garden outside the Shepherd Laboratories, University of Minnesota. Fig. 8(a) shows a top-down view of this area by stitching

(a) (b)

Fig. 8. To see if the pre-trained network is useful for different environments and different ground robots, we test with a garden area with a Clearpath Husky robot. (a) The ground appearance of this garden area is different from the cornfield, which we used to train the network. (b) We also use a different robot, Clearpath Husky. This ground robot is heavier than the robot in Fig. 4(b), thus it usually consumes more energy on the same ground.



(a) (b) (c)

Fig. 9. Testing the training result in a new environment with a new ground robot. (a) An aerial images of the test environment. (b) The energy-cost map by directly using the pre-trained network from the training set. Though we can still see the differences among classes, the energy-cost value is not accurate. (c) After a five-minute fine-tuning, the network outputs a better energy-cost map.

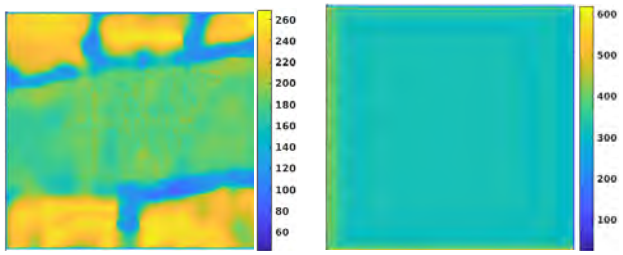
all the aerial images over this area. The total area is around $600m^2$. We also use a different ground robot, the Clearpath Husky as shown in Fig. 8(b), for this test. The measurements with this ground robot take on average $170J/m$ on the dirt road, and $160J/m$ on the concrete road. It takes $240J/m$ on the grass area, which is significantly higher than the other two terrain classes. Fig. 9(a) shows one of the aerial images from this area. Directly applying the network does not give good performance, though we can still see the energy-cost differences among terrain classes, as illustrated in Fig. 9(b).

To make the previous training result work for the new environment, we labeled four images from the test set to retrain the network. The retraining takes less than five minutes before the loss drops within 2% of ground truth value. The performance improves significantly, as shown in Fig. 9(c). The output of the network is consistent with our measurements. Therefore, our method can be generalized for various environments and ground robots with a fine-tuning (minutes), without the need of training from scratch (hours).

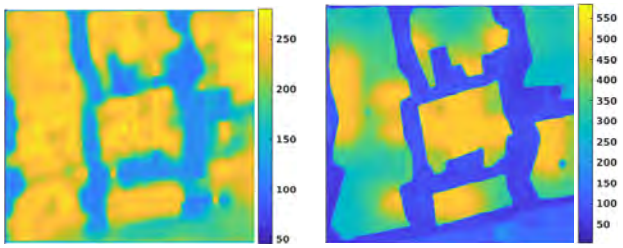
Remarks: If a terrain class costs more energy for one ground robot compared to other classes, it usually also costs more energy for another ground robot. We may not need the absolute values for finding the most energy-efficient terrain class to travel. However, we still need the map in other applications, such as planning the shortest path within a certain energy budget.

G. The Effects of Energy Measurements

We augment the sparse measurements with synthetic data to show that the network learns from both the images and the ground measurements, instead of memorizing energy cost values only based on the images. For this purpose, we change the values for the sparse ground energy-cost measurements while the network weights and the input images remain the same. We test whether the network outputs also change accordingly. In Fig. 10, we show examples where we set all the measurements to be the same value (Fig. 10(b)), or we multiply the measurements by 2 (Fig. 10(d)). The sparse measurements affect the network outputs, which indicates that the network learns from these measurements.



(a) (b)



(c) (d)

Fig. 10. Comparison of the network outputs when changing the sparse energy-cost measurements. The network weights and input images remain the same. The network output changes accordingly, which indicates that the network learns from both the images and the measurements. (b) The input energy-cost measurements are all set to be 400. The network outputs 400 for the whole map. (d) The input energy-cost measurements are multiplied by 2 compared to the measurements in Fig. 5(e). The values in the estimated map also increase.

V. Field Experiments

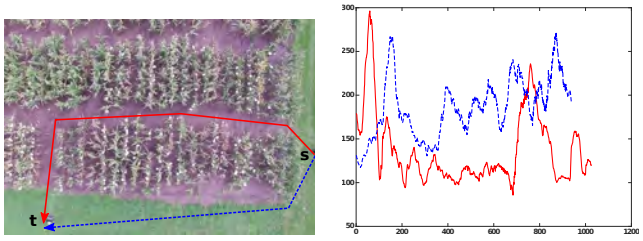
To demonstrate the practical application of our work, we use the energy-cost maps built by our method for path planning in the field. We compare the paths planned from the map with the shortest paths. We show that the ground robots consume less energy using the estimated the energy-cost maps.

We first test the result in the cornfield where we obtained the training images with the ground robot in Fig. 4(b). Fig. 11 shows an example of comparing the path planed using the map (red solid) and a shorter path (blue dashed). The length of our path is $28.8m$, and it takes $3534J$ of energy to follow it. The length of the shorter path is 23.4 meters, and it takes $3975J$ to follow it. Our method saves 12.4% in energy in this example.

We also test the map built by our method in the area shown in Fig. 9(a). To build the map, we fine-tuned the network trained with the data from the cornfield.

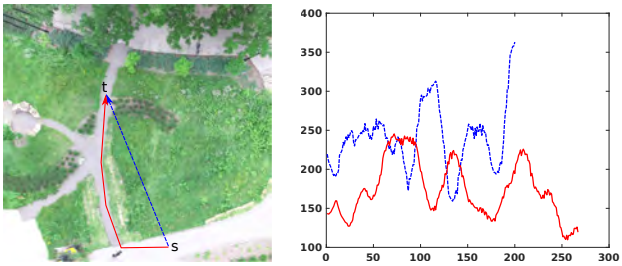
The red solid line in Fig. 12 shows an example of the planned path using our method. With the energy-cost map, the robot should travel on low-cost areas (concrete and dirt robot), instead of the grass area. We compare it with the shortest path, which is the straight, blue dashed line in Fig. 12(a). Fig. 12(b) plots the unit-distance energy consumption along the way. Our path takes $3423J$ energy in total, and the shortest path consumes $3983J$ energy. In this example, our method saves 16.3% energy.

Comparing the energy cost difference on different terrains in the experiments, ideally we can have an energy saving over 30%. However, the robot needs to turn when following the red solid paths, which consumes more energy compared to straight movements. Meanwhile, in Fig. 11(a), there is a height gap between the dirt road and grass areas. Our current method does not include the costs due to this gap. But we have shown that our current method still finds more efficient paths.



(a) (b)

Fig. 11. A path planning example in the cornfield using the energy-cost map built by our method. We compare the path with a shorter path. In this example, our path saves 12.4% energy. (a) The solid, red line is the trajectory planned using the map. The blue, dashed straight line a shorter path. (b) The unit-distance energy cost e along the trajectories. y-axis unit: J/m. x-axis: time steps. We record e at $20Hz$. The blue, dashed line corresponds to the blue, dashed trajectory in Fig. 11(a)



(a) (b)

Fig. 12. An example of path planning using the energy-cost map built by our method. We compare the path with the shortest path. In this example, our path saves 15% energy. (a) The solid, red line is the trajectory planned by our method. The blue, dashed straight line is the shortest path to the destination. (b) The unit-distance energy cost along the trajectories. The blue, dashed line corresponds to the blue, dashed trajectory in Fig. 12(a).

VI. Conclusion and Future Work

In this paper, we presented a learning method to build an energy cost map from aerial images and sparse ground robot measurements. We also showed that the trained network in one environment can be fine-tuned within minutes for a new environment with unknown terrain classes and robot. The learned network predicts energy-cost maps and enables us to plan energy-efficient paths.

In general, robot specific energy consumption also depends on robot states such as velocity and orientation. The energy-cost maps in this paper are predicted assuming that the robot is moving at a constant speed. In our future work, we would like to incorporate robot states and the terrain at the same time when building the maps. In this way we will be able to estimate the energy costs of more complex motions on terrains.

ACKNOWLEDGMENT

This work is funded in part by NSF grant #1525045, NSF grant #1617718, NSF grant #1849107, and MN State LCCMR program.

REFERENCES

[1] Anna Atramentov and Steven M LaValle. Efficient nearest neighbor searching for motion planning. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 632–637. IEEE, 2002.

[2] R Omar Chavez-Garcia, Jérôme Guzzi, Luca M Gambardella, and Alessandro Giusti. Learning ground traversability from simulations. *IEEE Robotics and Automation Letters*, 3(3):1695–1702, 2018.

[3] Jeffrey Delmerico, Elias Mueggler, Julia Nitsch, and Davide Scaramuzza. Active autonomous aerial exploration for ground robot path planning. *IEEE Robotics and Automation Letters*, 2(2):664–671, 2017.

[4] Mohamed Elbanhawi and Milan Simic. Sampling-based robot motion planning: A review. *Ieee access*, 2:56–77, 2014.

[5] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2015.

[6] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[7] Sven Koenig and Maxim Likhachev. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3):354–363, 2005.

[8] James J Kuffner Jr and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *ICRA*, volume 2, 2000.

[9] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[10] Shuang Liu and Dong Sun. Minimizing energy consumption of wheeled mobile robots via optimal motion planning. *IEEE/ASME Transactions on Mechatronics*, 19(2):401–411, 2014.

[11] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28, 2016.

[12] Yongguo Mei, Yung-Hsiang Lu, Y Charlie Hu, and CS George Lee. Energy-efficient motion planning for mobile robots. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4344–4349. IEEE, 2004.

[13] Patrick A Plonski, Pratap Tokekar, and Volkan Isler. Energy-efficient path planning for solar-powered mobile robots. *Journal of Field Robotics*, 30(4):583–601, 2013.

[14] Patrick A Plonski, Joshua Vander Hook, and Volkan Isler. Environment and solar map construction for solar-powered mobile systems. *IEEE Transactions on Robotics*, 32(1):70–82, 2016.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[16] Fabian Schilling, Xi Chen, John Folkesson, and Patric Jensfelt. Geometric and visual terrain classification for autonomous mobile navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2684. IEEE, 2017.

[17] Zheng Sun and John H Reif. On finding energy-minimizing paths on terrains. *IEEE Transactions on Robotics*, 21(1):102–114, 2005.

[18] Kshitij Tiwari, Xuesu Xiao, and Nak Young Chong. Estimating achievable range of ground robots operating on single battery discharge for operational efficacy amelioration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3991–3998. IEEE, 2018.

[19] Pratap Tokekar, Nikhil Karnad, and Volkan Isler. Energy-optimal trajectory planning for car-like robots. *Autonomous Robots*, 37(3):279–300, 2014.

[20] Minghan Wei and Volkan Isler. Air to ground collaboration for energy-efficient path planning for ground robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE.

[21] Minghan Wei and Volkan Isler. Energy-efficient path planning for ground robots by combining air and ground measurements. In *Conference on Robot Learning, 2019*.