2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

A Self-Supervised Near-to-Far Approach for Terrain-Adaptive Off-Road Autonomous Driving

Orighomisan Mayuku[1], Brian W. Surgenor[1], *Senior Member IEEE*, Joshua A. Marshall[2], *Senior Member IEEE*

***Abstract*— We introduce a self-supervised method for systematically choosing traversable terrain while autonomously navigating a vehicle to a goal position in an unknown off-road environment. Leveraging the color discriminant bias of off-road terrain types, and using images from a vehicle-mounted camera, we employ a viewpoint transformation that maintains the spatial layout of the terrain to cluster terrain types by color and register corresponding traversability features to guide future navigation decisions. As it navigates, our algorithm also generates training images for use in contemporary end-to-end navigation schemes. Our test results demonstrate the advantages of our approach over classical near-to-far approaches in off-road environments with unknown traversability characteristics, and highlight its fit to supervised semantic segmentation schemes that require foreknowledge of traversability characteristics for labeling, which are limited by insufficient data and suffer pixel-level class imbalance. We detail the techniques for clustering, feature registration, path planning and navigation; and demonstrate the method. Finally, we study the effectiveness of non-discretionary self-supervised data labeling.**

I. INTRODUCTION

In structured environments, well-paved and delineated roads mean that navigation decisions can be mainly guided by obstacle avoidance techniques, traffic signs, and regulations. However, in unstructured off-road environments containing combinations of mostly natural terrain types, navigation decisions are necessarily influenced by the choice of terrain type on which to drive, as measured by traversability features such as roughness and slip.

Like in urban driving, contemporary navigation techniques based on semantic segmentation using shape-discriminating convolutional neural networks (CNNs) have been applied to off-road environments, but with limited success [1], [2]. This reduced performance is partly due to limited datasets. While semantic segmentation datasets are generally limited compared to per-class datasets because of the pixel-level labeling constraints, the off-road datasets are particularly limited because of the remoteness of the environments for data collection. For example, the Freiburg Forest off-road dataset [1] contains only hundreds of images compared to tens of thousands in the Cityscapes urban dataset [3]. For context, the benchmark per-class dataset, ImageNet, comprises tens of millions of images. Furthermore, pixel-based classification exacerbates dataset class imbalance. For example, vegetation pixels typically dominate water puddles.

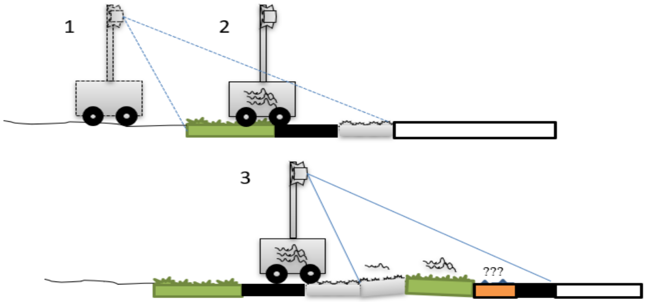

Fig. 1. A self-supervised near-to-far approach. Step 1: Terrain image is acquired and similar regions clustered by color; Step 2: Representative regions of each terrain type are driven over to acquire traversability features or costs and associate to color features; Step 3: Known traversal costs are used to plan a path through the next image. This figure is adapted from [5].

To improve segmentation performance, other modalities, such as depth, have been fused with color with modest performance improvements [1], [2]. A further complication is the reduced shape-based class distinctness which CNNs rely on. For example, vegetation is subjectively subdivided into “grass” and “traversable grass”, and individual clusters of one class are not unique in shape. This lack of class distinctness is less of a problem with uniform terrain samples [4]. Moreover, utilizing the segmented images for navigation presumes foreknowledge of traversability characteristics for each class, which is not guaranteed.

In terms of form, an image of a field of loose sand is not very different from one of red clay, but there is marked difference in terms of color. Vision-based near-to-far methods take advantage of this color dependence to associate color and traversability features for off-road navigation [5]. In a self-supervised scheme (Fig. 1), the traversability features are acquired and associated with the color features online. Current methods require some supervision for feature registration and are not end-to-end because the classical learning models used require feature extraction. An obvious challenge is the variations in chroma and illumination for a particular terrain type. We tackle this challenge by using classical clustering techniques.

In the work presented in this paper, we present a self-supervised method for systematically navigating an exploratory skid-steer robot to a goal position by acquiring and robustly associating traversability and color features online for an unknown off-road environment with reasonably flat terrain. We show how this method fits into contemporary

[1] O. Mayuku and B. W. Surgenor are with the Department of Mechanical & Materials Engineering and the Ingenuity Labs Research Institute, Queen’s University, Kingston, ON K7L 3N6 `o.mayuku@queensu.ca`, `surgenor@queensu.ca`

[2] J. A. Marshall is with the Department of Electrical & Computer Engineering and the Ingenuity Labs Research Institute, Queen’s University, Kingston, ON K7L 3N6 `joshua.marshall@queensu.ca`

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

end-to-end schemes that use semantic segmentation; and we critically evaluate its effectiveness. Our contribution includes using a transformed representation of the terrain to take advantage of the color bias of terrain features, and a new systematic self-supervised feature registration approach.

II. RELATED WORK

Traversability features, such as roughness and slip, have long been characterized empirically based on terramechanics [6] or estimated with sensors [7]. Supervised machine learning tools have also been used to classify these features [8] along with simple NNs [9]. In [10], some of these techniques were also reviewed.

A. *CNN-Based Off-road Navigation Methods*

Due to the limited results obtained from color-only semantic segmentation [1], [2], multimodal fusion has been explored. In [1], multispectral fusion with near-infrared (NIR) channel data was explored; and while 3D LiDAR fusion gives improved results, the size of the resulting network reduces scope for online use [2]. Stereo depth maps are one option, but they are particularly noisy and require further processing. Estimating depth with a CNN is another option [11], but this complicates the prediction pipeline by adding a preprocessing step; moreover, in [2], RGB-3D data fusion did not give good results for rough terrain. The best fusion scheme is determined by experimentation [12]. Transfer learning methods have also been used [13], [14]. One main challenge with using separate sensors online is the need for geometric matching of the different modalities.

Specifically related to off-road terrains, in [15], trail direction was inferred from aerial images, and, in [4], labeled images of terrain were obtained from a walking robot with small footprint. Although the robot's limited footprint meant that pixel labels were significantly sparse, the uniformity of the terrain facilitated segmentation. More recently, CNN-based reinforcement learning has been used to iteratively learn paths through rough off-road terrain [16], however, this scheme requires multiple learning runs and will only work for the learned terrain. The overwhelming challenge with these CNN-based methods is the significant resources required for data collection, curation, and labeling.

B. *Vision-based Near-to-Far Methods*

Proposed during the DARPA LAGR program [17], these methods associate proprioceptive sensor data to exterioceptive sensor data to extend the predictive range of the robot [18], [19], [20], [21]. In one scheme, a supervisory vibration classifier was used to label the training data for a color classifier to predict the traversability of future terrains from images [5], [22]. However, the online data collection scheme did not systematically identify and deliberately traverse uniform terrain sections for data collection, thus, feature registration still entails some supervision. Similarly, a probabilistic approach without training was presented in [23], however, the data collection trajectories were specified, and variations in terrain type and structure were limited. Due to their noisy nature, disparity maps cannot be used to predict roughness at a practical resolution, and slip cannot be reliably estimated from vision, either; so, acquiring traversal data requires driving over the terrain. Thus, in another scheme, separate models were trained with provided semantic labels for color and traversability features respectively, then co-trained bi-directionally [8] in a classification scheme [9].

However, these classical classification approaches do not suit contemporary end-to-end navigation schemes that are based on semantic segmentation using CNNs. Unlike [5], [8], our method autonomously chooses uniform terrain patches for self-supervised feature registration while autonomously navigating the robot to a goal in an end-to-end scheme. As it progresses towards the goal, our method collects images and labels image features by traversal cost in the image plane to train a CNN for robust semantic segmentation.

III. SELF-SUPERVISED NEAR-TO-FAR APPROACH

First, we use a perspective transformation to maintain the 2D spatial distribution of terrain features to enable clustering of similar terrain classes based on visual features, then we present a traversability metric for path planning decisions and detail our approach to navigating to a specified goal position.

A. *Visual Features and Clustering*

Clustering is fundamental in unsupervised color-texture approaches [17]; because of its robustness to noise, we used DBSCAN [24] on consistent color features. The required adjacency map was built by using the SLIC superpixel algorithm [25]; chosen because of its ease of tuning. To compute color difference (ΔE), these algorithms require a perceptually uniform color space. Traditionally the L*a*b* space was used–where L^* represents lightness, and the chromatic features represent the green-red and blue-yellow components; and ΔE was based on a Euclidean distance metric (CIE76). For a cluster of pixels, the mean or median intensities are used. However, color models are continually improved: we used the J′a′b′ space from the CAM16-UCS [26] model based on CIECAM02. For two patches from an area of grass terrain at different illumination levels, the improvements in performance are shown in Fig. 2. Note that, in some cases—e.g., for black tar vs granite, L*a*b* can give

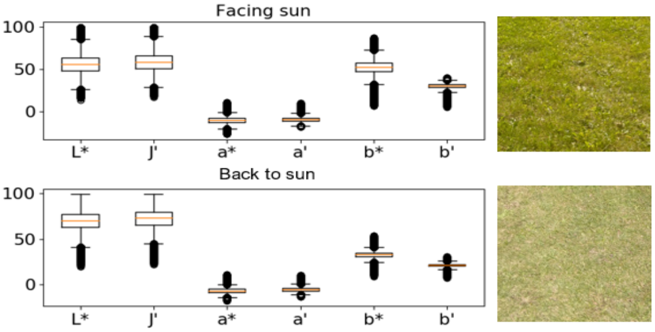

Fig. 2. Comparison of L*a*b* and J′a′b′ color spaces for two patches of outdoor grass. The individual J′a′b′ features show comparatively less spread, with a smaller chromatic ΔE of 9.8 versus 20 for L*a*b*.

better results. While clustering similar classes, neglecting the achromatic J' helps reduce the effects of illumination by correctly giving a smaller ΔE. Thus, we use

$$\Delta E = \sqrt{(a'_1 - a'_2)^2 + (b'_1 - b'_2)^2}. \tag{1}$$

While previous approaches have clustered superpixels in the 2D image plane [8], this plane does not preserve the geometric layout of the terrain for self-supervised feature registration and requires external depth correlation. In our approach, to maintain the geometric layout of the terrain, we employ an empirical approach to perspective transformation. Accounting for the camera field of view (FOV), a fixed region of interest (ROI) in the image is transformed to the "top-down" view by using least squares and RANSAC. The ROI represents a rectangle in top-down view as in Fig. 3. If (u, v) and (U, V) represent the vertices in the transformed image and ROI respectively, then M maps

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = M \begin{bmatrix} U_i \\ V_i \\ 1 \end{bmatrix}, \tag{2}$$

where $i = 1, \ldots, 4$. Thus, for any pixel (u, v) in the transformed image (Tr), the intensity can be approximated from the ROI as

$$\mathrm{Tr}(u, v) = \mathrm{ROI}(f_x(u, v), f_y(u, v)). \tag{3}$$

Bilinear interpolation is used to convert continuous values to discrete ones in image space. The homogeneous coordinates have been scaled to image coordinates; f_x and f_y parameters are in M^{-1}. The position of the fixed ROI depends on the camera height and angle. While the ROI is invariant to yaw angle, it is affected by pitch and roll of the robot frame which holds the camera; but this variance is not unique to this scheme [27], and similar mitigating methods apply.

For the transformed image, smaller real-world sizes approximate terrain undulations better, and interpolation effects become more pronounced with bigger ROI sizes. The transformed image is scaled to real-world size in (3). This scaling is important because it determines the resolution of waypoints from the path planner. Smaller sizes mean faster processing, with less noise; but increasingly distant waypoints with less-smooth path following.

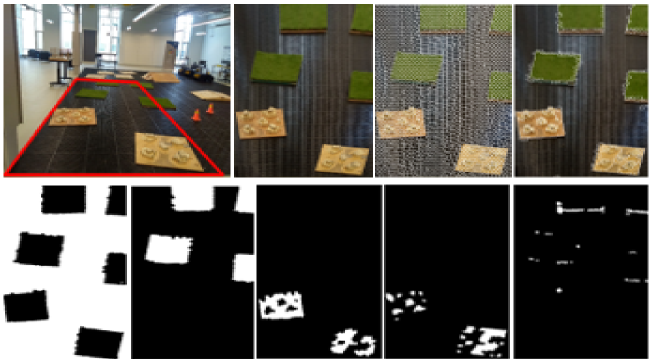

Fig. 3. The image processing pipeline (L-R): Top: Source Image and ROI; Transformed image; Superpixel clusters; DBSCAN clusters. Bottom: Uniquely labelled classes, noisy classes (e.g. bottom right) are discarded morphologically and in a future processing step. To collect traversability data, the robot drives over the largest cluster per class. Note how the darker tyre marks on the wood are separately clustered.

In clustering the ROI into superpixels, SLIC considers positional distance (d_s) and ΔE. Consider N number of pixels, and k number of superpixels, for uniform-sized superpixels, the grid interval is: $S = \sqrt{N/K}$. And for pixel (i, j), the combined SLIC distance metric D is

$$d_s = \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} \tag{4}$$

$$D = \sqrt{(\Delta E)^2 + (d_s/S)^2 p^2}, \tag{5}$$

where p specifies the relative priority of color or distance.

The resulting superpixels are clustered by using DBSCAN with an adjacency matrix and an empirical ΔE threshold (E_{DB}). The resulting labeled clusters are not uniquely numbered. Thus, as a final step, clusters are compared and renumbered uniquely if they meet a less-strict ΔE threshold (E_I)–without regards for proximity (Fig. 3, bottom). After the first run, a feature list of per-class mean color features F_v is stored as

$$f_i = [J', a', b']^T \tag{6}$$

$$F_v = \left[f_i, \ldots, f_n\right]. \tag{7}$$

where n is the number of known unique clusters. Subsequently, given a new clustered image, to identify previously-seen and unseen classes, the ΔE is calculated for class j in the new clustered image with respect to F_v based on an empirical color threshold E_T, and stored in list U as

$$\Delta E_{f_i,f_j} = \begin{cases} \Delta E_{f_i,f_j} & \Delta E_{f_i,f_j} < E_T \\ 0 & \Delta E_{f_i,f_j} \geq E_T \end{cases} \tag{8}$$

$$U = [\Delta E_{f_i,f_j}, \ldots, \Delta E_{f_n,f_j}] \tag{9}$$

and, in F_v, the new class number m for this class j is

$$U = 0, \text{ if } \Delta E_{f_i,f_j} \geq E_T, \quad \forall i = 1, \ldots, n \tag{10}$$

$$m = \begin{cases} \arg\min_{i \in n}(U), & U \neq 0 \quad \text{(11a)} \\ n + 1, & U = 0. \quad \text{(11b)} \end{cases}$$

Note that F_v is updated accordingly.

The effect of E_I is illustrated in Fig. 4. The color difference thresholds were determined empirically by comparing similar color pairs as in Fig. 2. Outdoor grass was used as a benchmark because of its wide variations in color and illumination. Another benchmark color pair is black tiles vs

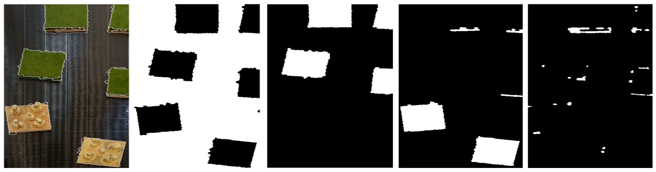

Fig. 4. Clustering results for the indoor terrain in Fig. 3 with $E_I = 17$. The threshold is not sensitive enough to identify the much darker sections on the rough wood terrain caused by tyre wear.

gravel, which show chromatic similarity. The thresholds are slackened as cluster size increases due to increased noise. A stricter threshold increases the likelihood of false negatives. Table I shows a list of our tuned parameter ranges.

B. Traversability Metric

Traversablity is usually measured in terms of roughness and slip. While a lot of work has been done in modeling slip [10], it remains difficult to model. Thus, we limit our work to roughness, indirectly estimated by using an IMU [8] in the time domain. On rough terrain, vibrations are tri-axial, thus we combine the IMU's linear acceleration readings, $a = [a_x, a_y]$. Table II shows sample data for terrain types in Fig. 3. The narrow range necessitates scaling to magnify small differences for the path planner. Thus, for terrain i with N samples, the traversability feature list is F_T such that

$$f_{T,i} = e^{8\left(\frac{1}{N}\sum \|a\|_2\right)} \tag{12}$$

$$F_T = \left[f_{T,i}, \ldots, f_{T,n}\right], \tag{13}$$

where n is the number of known unique clusters. The practical feature matching process is discussed in the System Overview section. Thus, the running list of features is

$$F = [F_V, F_T]^T. \tag{14}$$

C. Path Planning and Control

Our approach makes use of established methods for path planning and control. A common strategy is to choose the best option from reachable candidate paths based on a cost metric. But our image-based algorithm fits a grid-based approach that suits the excellent manoeuvrability of our exploratory skid steer robot. While grid-based any-angle path planners are an attractive option because of node edge-independence, optimal any angle path planners often require clearly designated obstacle and free nodes [28]. But, in our case, in addition to obstacle nodes, traverseability is treated as a hierarchy, with nodes ranked by difficulty of traversal. Thus, we simply approach the problem with a naïve A* path planner using the optimistic heuristic from [29]. The nodes are the pixels with traversal costs in (13).

We use a feedback linearized (FBL) path-following controller based on a simplified kinematic model. The errors are based on instantaneous closest path points. The reader is referred to [30], which we emulated to suit our purposes.

TABLE I

LIST OF TUNED IMAGE PROCESSING PARAMETERS.

Parameter	k^1	p	E_{DB}	E_I	E_T
Value	10 to 25	< 35	< 4	< 8	< 13

[1] Average pixels per superpixel.

TABLE II

TRAVERSAL DATA FOR OUR INDOOR TERRAIN AT 0.5 M/S.

Terrain	Grass	Rough wood	Granite	Rubber mat
$\frac{1}{N}\sum \|a\|_2$ [1]	0.683	2.723	1.262	0.948
$\frac{1}{N}\sum \|J\|_2$ [2]	1.860	12.114	5.526	5.209

[1] By acceleration, m/s^2. [2] Jerk, $J = [J_x, J_y]$, m/s^3.

D. Navigation and Localization

As in Fig. 5, the start position of the robot is the origin of its global map. The fixed position of the transformed ROI with respect to the robot $(x_{r,i,0}, y_{r,i,0})$ is determined by calibration. Thus, any waypoint in image coordinates (u, v), can be transformed to robot and map coordinates S_r and S_m, such that

$$S = \begin{bmatrix} x_{r,i,0} & y_{r,i,0} \end{bmatrix}^T \tag{15}$$

$$S_r = \begin{bmatrix} x_{r,i} & y_{r,i} & 1 \end{bmatrix}^T = \begin{bmatrix} I & S \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -v \\ -u \end{bmatrix} \tag{16}$$

$$S_{xy} = \begin{bmatrix} R_{m,r,x} & R_{m,r,y} \end{bmatrix}^T \tag{17}$$

$$S_m = \begin{bmatrix} x_{m,i} & y_{m,i} & 1 \end{bmatrix}^T = \begin{bmatrix} R(\theta) & S_{xy} \\ 0 & 1 \end{bmatrix} S_r, \tag{18}$$

where $R(.)$ is the rotation matrix. As in (18), the *UTM-Map* transformation is found using $(R(\phi), R_{utm,g})$. Thus, the goal is specified in Cartesian coordinates; or in GPS coordinates that are transformed to Cartesian coordinates by using the *UTM-Map* transformation.

IV. EXPERIMENTAL SYSTEM OVERVIEW

Our test platform was a Clearpath Husky A200 unmanned ground vehicle (UGV), a skid-steer robot with a maximum speed of 1 m/s. Husky was equipped with wheel encoders and a calibrated triple-axis LORD Microstrain 3DM-GX5-25 IMU comprising a magnetometer, gyroscope and accelerometer. Images were collected with an 8 MP monocamera mounted on Husky at a height of 0.5 to 1.2 m. To reduce light reflection from reflective classes e.g. gravel, the camera was set at about 0° to 10° from the horizontal towards the ground. Outdoors, a Swift GPS unit was used with real-time kinematic (RTK) corrections. The GPS, wheel encoder and IMU sensor data are fused using an extended Kalman Filter hosted on a Intel i7 Lenovo computer which runs the Robot Operating System (ROS) on Ubuntu for interacting with Husky. A Vicon motion camera tracker system was used for localization during indoor testing.

Our indoor prototyping terrain is a square flat area of sides 8.6 m with a base synthetic black mat. Three reconfigurable terrain elements are used: (1) wood with distributed wooden

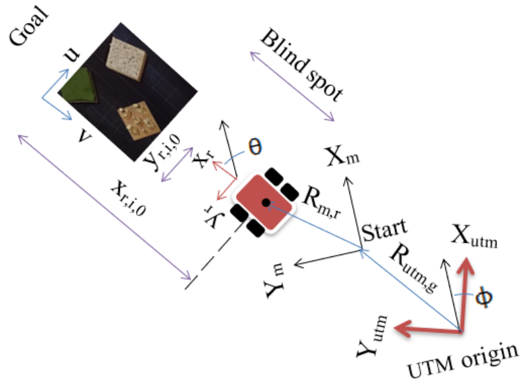

Fig. 5. Localization, navigation and path following strategy.

protrusions (2) gravel on a wooden base and (3) synthetic grass on a wooden base. Each unit is square with 0.9 m sides. These four terrain types represent some of the typical classes found in an off-road environment and exhibit representative color differences. The fixed ROI, covers a square area of sides 3.2 m as empirically chosen to reduce interpolation effects. The indoor terrain and hardware setup are shown in Fig. 6. Three scenarios summarize the approach: (1) class generation; (2) class identification; and, (3) novelty detection.

A. *Scenario 1: Class Generation*

In Scenario 1, at start up, the feature list is empty, the algorithm identifies the per-class representative clusters by size. For each representative cluster that meets a size and solidity threshold for robot traversal to collect data, the algorithm fits a bounding contour [31], and then fits a minimum area bounding box around the contour. Using zero costs, paths are planned so that the robot drives longitudinally through the center of each box for feature registration before proceeding to the interim goal position (Fig. 7, left top-bottom pair). Note that the bounding contour for the black tiles is the whole image, thus, a region near the robot is sampled. Also, note that the path sequence follows the terrain class numbers; this sequence is important for feature registration (14). The image waypoints are transformed to map coordinates for path following.

B. *Scenario 2: Class Identification*

In Scenario 2 the robot has previously encountered all the terrain classes, so it uses the running feature list to generate a labeled traversabilty cost image then plans a path to the interim goal as in Fig. 7 (middle top-bottom pair).

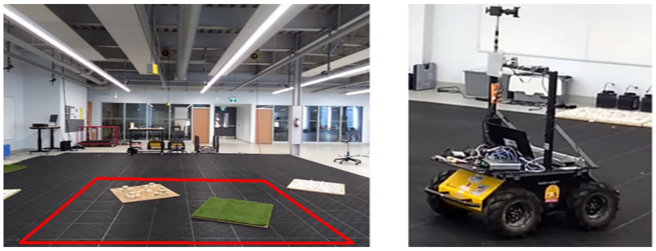

Fig. 6. Left: Indoor terrain with reconfigurable terrain elements. Right: Test platform, Clearpath Husky A200 UGV, with sensors and mounts.

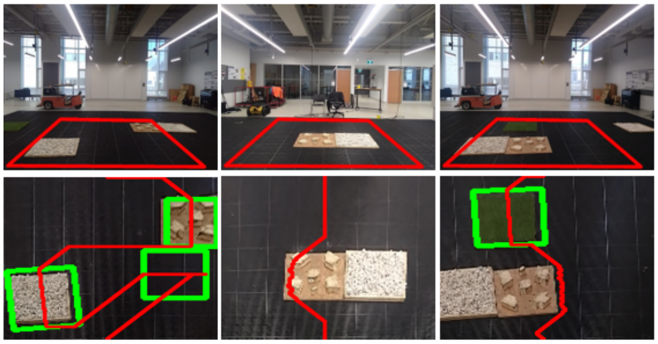

Fig. 7. The three test scenarios (L-R): Top: Source image with ROI. Bottom: Transformed ROI with paths from path planner.

C. *Scenario 3: Novelty Detection*

Finally, in Scenario 3 (Fig. 7, right top-bottom pair), the robot identifies the new grass cluster, and fits a bounding box around it to which a path is planned using known costs. Lastly, a path is planned to the interim goal. In all cases, the robot assumes a direct heading to the specified final goal position; the interim goal position is the line of intersection with the image. In our indoor tests, the robot performed Scenarios 1 and 3 in a forward run and Scenario 2 in a backward run; all at 0.5 m/s. Between each scenario, the terrain elements were rearranged.

V. EXPERIMENTAL RESULTS

Because of the grid dependence, the paths have sudden breaks and can be choppy. To handle these breaks, the paths were broken into smooth sections in post-processing such that at the end of each section, the vehicle rotates in place to the start pose of the next section before continuing. This strategy takes advantage of the manoeuvrability of the skid-steer robot particularly as speed is not critical in this context. Nonetheless, the basic FBL controller still struggles with the sharp turns as seen in Fig. 8, although this is not the focus of this work. Also, the paths tend to closely follow high cost terrains, meaning that the robot will partially drive over the edges of such terrains. This can be solved by adding allowance regions around high cost clusters.

Furthermore, classes that are too small to be driven over to collect data (Fig. 3, bottom right and second right) were treated as identical to low cost regions, otherwise they have unrealistic effects on path planning. For example, the grass section—which has the lowest cost—is walled off by its high cost wooden borders. However, the implication is that paths can be planned though unrealistic crevices such as between rough patches. Notice how the path is planned through the tyre burn marks in the Fig. 7 (middle) but not in the reverse run in Fig. 7 (right). This is because the tyre burn marks were identified as a continuous section as in Fig. 3 (bottom second right) in the first case but as a discontinuous noisy patch in the second case, all of unknown traversability. Smaller noisy clusters were removed morphologically.

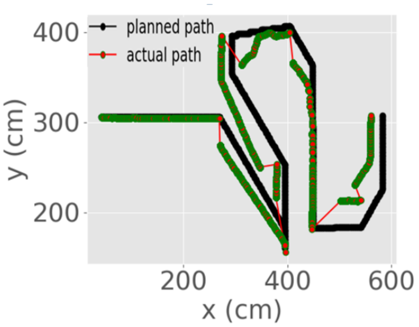

Fig. 8. Scenario 1: Class generation. The paths in Fig. 7, left, have been transformed to map coordinates using (18) and a straight heading from start across the blind spot has been concatenated. The FBL controller does not track the sharp turns as well as the straight paths.

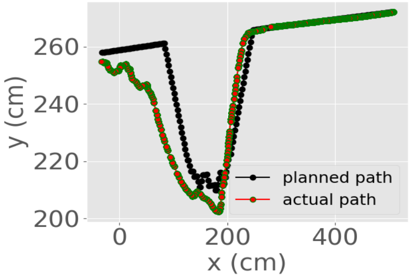

Fig. 9. Scenario 2: Class identification. In this reverse run, the path has fewer sharp turns because it comprises only two paths: a straight heading across the blind spot and the paths from Fig. 7 (middle) all in map coordinates. The path following error is more pronounced during turning due to skidding and controller undercutting.

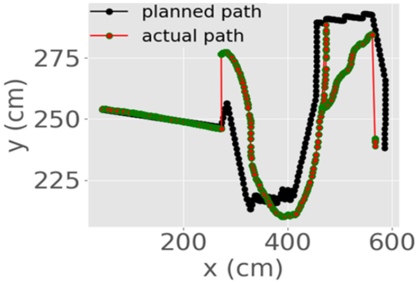

Fig. 10. Scenario 3: Novelty detection. In this case, the significant path following error at the end is caused by unintended skidding as the vehicle turns and drags the wooden terrain element along Fig. 7 (right). Keep in mind that the point being tracked is the front midpoint of the vehicle.

The advantage of a naive grid-dependent path planner is that it helps maintain a sense of heading with respect to the specified goal in the unmapped environment. The main alternative, that uses path proposals and makes decisions based on path costs, benefits from defined path boundaries to maintain a sense of heading. Currently, the robot naively drives a direct path across the blind spot; this, along with incorporating obstacle avoidance and robot workspace-aware path planning, is the subject of future work.

In Fig. 8 to Fig. 10, the controller path following errors are magnified by skidding as the robot drags the freely-moving indoor terrain elements along as it spins to turn. This was not intended. In addition, the controller errors are based on the instantaneous closest point to the vehicle which means sharp turns may be undercut—not diligently followed.

Note that the system requires systematic tuning of color thresholds to achieve robustness because it is meant for use in the absence of labeled data—which is never adequate—or for terrain comprising classes with unknown traversal costs. However, taking the classical approach of self-supervision as a tool for labeling data for training [4], [5], [8], without simultaneously navigating to a goal, then robustness is not critical. Strict color thresholds will increase false negatives, which is preferred because labels are based on traversal costs; e.g., two grass clusters identified as different classes by color will still have the same range of traversal cost. However, non-unique clusters reduce navigation efficiency (Fig. 11).

VI. Self-Supervised Labeling for Semantic Segmentation

To leverage the robustness and speed of supervised end-to-end schemes, a semantic segmentation network can be trained on the database of transformed ROI—cost image pairs. In addition to a generic obstacle class, clusters that do not meet the size limit for data collection can be treated as an unknown class; and the cost image thresholded into unique class labels in steps of 0.5 units (Table II). Fig. 11 shows examples of clustered images for representative terrains. While the non-discretionary labeling of clusters results in naively labeled pixels, human-labeled data also struggles in these environments as seen in [2] where fusion with high resolution LiDAR did not give good results for rough terrain classification. From Fig. 11 (upper figure), the non-uniqueness in shape across classes due to the natural random distribution of terrain classes in off-road environments becomes evident. Similar problems have been tackled in the field of remote sensing whereby texture features are relied on to segment 2D color and monochrome images. If the ROI is well-sized, and the distortion caused by the transformation is considered consistent, then such texture features are also relevant here since CNNs are also sensitive to texture [32].

VII. Conclusion

We have presented and implemented a self-supervised method for systematically navigating an exploratory skid-steer robot to a goal position by acquiring and associating traversability and color features online for an unknown off-road environment with reasonably flat terrain. We have shown how this method fits into contemporary end-to-end schemes that use semantic segmentation. Our contribution includes using a transformed real-world 2D representation of the terrain to take advantage of the color bias of terrain features, and a new systematic self-supervised feature registration approach. Future work includes incorporating slip, obstacle avoidance, ground-air vehicle teaming with aerial images, speed adaptation, and further testing in outdoor environments with more variability in the encountered terrains.

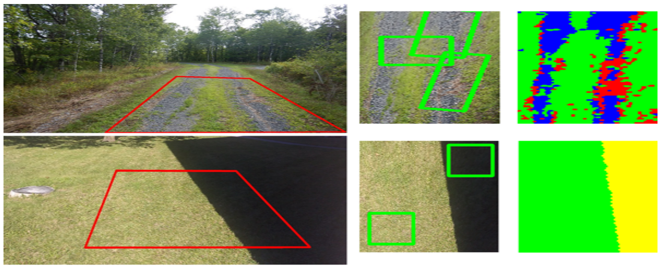

Fig. 11. The proposed method on sample off-road terrains using fixed parameters (Table I). Left: Terrain with sample ROI. Middle: Transformed ROI with regions for feature registration. Right: Clusters identified as unique by algorithm. For the bigger clusters a representative section was sampled by the algorithm for feature registration. The lower figure shows the effect of illumination: although the terrain is all grass, a structure casts a shadow on a section, and the algorithm clusters it separately. But since labels are based on traversal costs, both sections will have the same label after feature registration. In the upper figure, terrain types are uniquely identified.

REFERENCES

[1] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, ”Deep multispectral semantic scene understanding of forested environments using multimodal fusion,” in *2016 International Symposium on Experimental Robotics*, Nakamura Y., Khatib O., Venture G. Eds, ISER 2016. Springer Proceedings in Advanced Robotics, vol 1. Springer, Cham.

[2] D.-K. Kim, D. Maturana, M. Uenoyama, and S. Scherer, ”Season invariant semantic segmentation with a deep multimodal network,” in *Field and Service Robotics*, Springer Proceedings in Advanced Robotics, 2018, vol. 5, pp. 255 – 270.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, “Where should I walk? Predicting terrain properties from images via self-supervised learning, in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, April 2019.

[5] C. A. Brooks and K. Iagnemma, “Self-supervised terrain classification for planetary surface exploration rovers,” *Journal of Field Robotics*, vol. 29, no. 3, pp. 445–468, 2012.

[6] M. Bekker, *Introduction to Terrain-Vehicle Systems*. The University of Michigan Press, 1969.

[7] A. Angelova, L. Matthies, D. Helmick, and P. Perona, ”Learning and prediction of slip from visual information,” *Journal of Field Robotics*, vol. 24, no.3, pp. 205–231, 2007

[8] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, “Autonomous terrain classification with co- and self-training approach,” *IEEE Robotics and Automation Letters* vol. 1, no. 2, pp. 814–819, Jul. 2016.

[9] E. M. DuPont, C. A. Moore, E. G. Collins, and E. Coyle, “Frequency response method for terrain classification in autonomous ground vehicles,” *Autonomous Robots*, vol. 24, no. 4, pp. 337–347, 2008.

[10] P. Papadakis, ”Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1373–1385, 2013.

[11] F. Liu, C. Shen, and G. Lin, ”Deep convolutional neural fields for depth estimation from a single image,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 5162-5170.

[12] D. Feng et al., ”Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges,” in *IEEE Transactions on Intelligent Transportation Systems*, ’02, 2020, pp. 1-20.

[13] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, ” Semantic segmentation with transfer learning for off-road autonomous driving,” *Sensors*, vol. 19, no. 11, 2577, 2019.

[14] C. J. Holder, T. Breckon, and X. Wei, “From on-road to off: Transfer learning within a deep convolutional neural network for segmentation and classification of off-road scenes,” in *Proc. European Conference on Computer Vision*, 2016, pp. 149-162.

[15] A. Giusti et al., ”A machine learning approach to visual perception of forest trails for mobile robots,” in *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661-667, July 2016

[16] T. Manderson, S. Wapnick, D. Meger, and G. Dudek, ”Learning to drive off road on smooth terrain in unstructured environments using an on-board camera and sparse aerial images,” in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*, June 2020

[17] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Sundaresan and A. Sola, “Mapping, navigation, and learning for offRoad traversal,” *Journal of Field Robotics*, 2009, vol.26, pp. 88-113.

[18] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, “Towards learned traversability for robot navigation: from underfoot to the far field,” *Journal of Field Robotics*, vol. 23, no. 11–12, p.1005-1017, 2006

[19] A. N. Erkan, R. Hadsell, P. Sermanet, J. Ben, U. Muller and Y. LeCun, ”Adaptive long range vision in unstructured terrain,” in *Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 2421-2426

[20] R. Hadsell et al., ”Online learning for offroad robots: Using spatial label propagation to learn long-range traversability,” *Robotics: Science and Systems*, Atlanta, Georgia, June 2007.

[21] M. J. Procopio, J. Mulligan and G. Grudic, ”Long-Term learning using multiple models for outdoor autonomous robot navigation,” in *Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 3158-3165

[22] M. Bajracharya, A. Howard, L. Matthies, B. Tang, and M. Turmon, ”Autonomous off-road navigation with end-to-end learning for the lagr program,” *Journal of Field Robotics*, vol. 26, no.1, pp. 3–25, 2009.

[23] A. Krebs, C. Pradalier, and R. Siegwart, “Adaptive rover behaviour based on online empirical evaluation: Rover-terrain interaction and near-to-far learning”, *Journal of Field Robotics*, vol. 27, no. 2, pp. 158–180, 2010.

[24] M. Ester, H-P., Kriegel, J., Sander, X., Xu, ”A density-based algorithm for discovering clusters in large spatial databases with noise” in *Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, 1996, pp. 226–231.

[25] Achanta et al., SLIC Superpixels, EPFL Technical Report 149300, June 2010.

[26] C. Li, Z. Li, Z. Wang, Y. Xu, M. R. Luo, G. Cui, M. Melgosa, M. Brill, M. Pointer, ”Comprehensive color solutions: CAM16, CAT16 and CAM16-UCS,” *Color Research and Application*, vol. 42, pp. 703-718, 2017

[27] M. Santoro, G. AlRegib and Y. Altunbasak, ”Misalignment correction for depth estimation using stereoscopic 3-D cameras,” in *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, Banff, AB, 2012, pp. 19-24

[28] D. D. Harabor, A. Grastien, ”An Optimal any-angle pathfinding algorithm”, in *Proc. 23rd International Conference on Automated Planning and Scheduling*, 2013

[29] H. Choset et al., Principles of robot motion: Theory, algorithms, and implementations, MIT Press. 2005. ISBN: 9780262255912

[30] J. A. Marshall, T. D. Barfoot, and J. Larsson. Autonomous underground tramming for center-articulated vehicles. Field and Service Robotics of the Journal of Field Robotics, vol. 25, no. 6-7, pp. 400-421, June-July 2008

[31] S. Suzuki, K.A. be, ”Topological structural analysis of digitized binary images by border following,” *Comput. Vis. Graph Image Process* vol. 30, pp: 32–46, 1985

[32] R. Geirhos et al., “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness,” in *Proc. International Conference on Learning Representations*, New Orleans, USA, 2019.