Geometric and Visual Terrain Classification for Autonomous Mobile Navigation

Fabian Schilling, Xi Chen, John Folkesson and Patric Jensfelt

***Abstract*— In this paper, we present a multi-sensory terrain classification algorithm with a generalized terrain representation using semantic and geometric features. We compute geometric features from lidar point clouds and extract pixel-wise semantic labels from a fully convolutional network that is trained using a dataset with a strong focus on urban navigation. We use data augmentation to overcome the biases of the original dataset and apply transfer learning to adapt the model to new semantic labels in off-road environments. Finally, we fuse the visual and geometric features using a random forest to classify the terrain traversability into three classes: *safe*, *risky* and *obstacle*.**

We implement the algorithm on our four-wheeled robot and test it in novel environments including both urban and off-road scenes which are distinct from the training environments and under summer and winter conditions. We provide experimental result to show that our algorithm can perform accurate and fast prediction of terrain traversability in a mixture of environments with a small set of training data.

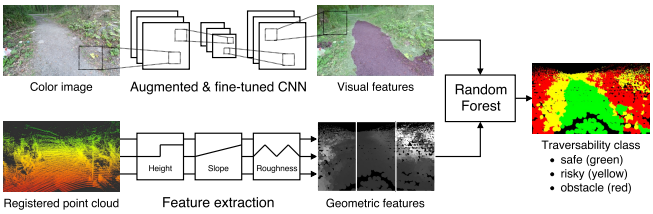


Fig. 1: Our terrain classification algorithm consists of two complementary processing pipelines. Visual features are extracted from color images using a fully convolutional network trained with data augmentation and fine-tuned to our target environment. Geometric features are computed efficiently from registered point clouds. Our system generates fast and accurate traversability estimates in novel environments by training a random forest on small amounts of labeled data.

I. INTRODUCTION

Terrain traversability classification is important for autonomous ground robots to safely navigate in real world environments. It is a highly challenging task as the robot has to deal with a variety of different terrain types that have distinct physical properties. Most approaches concentrate on specific types of terrain in either urban environments or off-road environments since both typically underlie different traversability assumptions. The terrain classification literature follows three main directions: an assessment based on scene geometry only, scene appearance only, and approaches that combine the two.

The geometry-based approaches rely on the reconstruction of the 3D environment using a lidar or stereo camera. The terrain traversability is estimated using the shape of the terrain surface [1], [2]. However, the geometry-based approach suffers from poor terrain estimates on deformable surfaces such as long grass and snow where the true ground is covered and undetectable from a distance.

The appearance-based approaches pose terrain classification as an image classification or semantic segmentation problem. The material of the terrain surface is categorized visually into a discrete set of classes [3], [4]. However, in many cases, knowing only the terrain material is not sufficient to decide the underlying traversability. For instance, an ice-covered road can be driven on carefully when it is flat but becomes non-drivable when it is sloped.

To overcome the limitations of using only the geometry-based or appearance-based approach, the intermediate feature representations can be fused together into a final traversability assessment with off-the-shelf learning methods such as random forests or support vector machines.

In our work, we focus on mixed terrain including both urban and off-road environments. We present a robust terrain classification algorithm that relies on transfer learning and can be applied to novel environments from which there is limited training data. As in [5] we move away from a binary classification of space into freespace and obstacle. We use three terrain traversability classes: *safe*, *risky*, and *obstacle*. The motivation being that there is no clear boundary between *safe* and *obstacle* in complex environments. Besides not being quit as safe to drive on, driving on *risky* terrain often causes navigation problems due to faulty point cloud registrations or perceptive failures due to motion blur. By anticipating these problems, we can apply special strategies such as slowing down, changes in motion primitives, or asking a human operator for assistance.

Our terrain classification algorithm relies on two complementary processing steps. We compute efficient geometric features from a lidar point cloud and train a fully convolutional network to perform pixel-wise semantic segmentation from the camera image. The geometric features and pixel-wise semantic predictions are then fused into a final terrain traversability class using a random forest classifier. We train our algorithm using a small set of labeled data and evaluate it on novel scenes. The overall process is outlined in Fig. 1

The contributions of this paper are i) a generalized terrain representation using geometric features with semantic labels and ii) use of transfer learning to adapt to novel environments and to introduce new semantic labels, iii) methods for data augmentation to overcome biases in the data, iv) an evalua-

tion on real-world data and v) a practical implementation of the approach.

The remainder of the paper is organized as follows: Sec. II presents relevant previous work. Sec. III describes the specific problem we deal with, the environment we operate in, and how we define the terrain classes we use. Sec. IV describes both our visual and geometric processing pipelines and their fusion into a final traversability class. Sec. V describes the experimental setup and Sec. VI presents the experimental results and analyzes the performance of our proposed method. Sec. VII presents our conclusions and possible directions of future work.

II. RELATED WORK

A comprehensive survey of terrain traversability analysis methods is presented in [6]. The authors categorize methods using exteroceptive sensory data processing as either geometry-based or appearance-based. For geometry-based approaches, [7] apply a fuzzy classifier to estimate terrain from a digital terrain model (DTM). [1] and [2] construct a 2D traversability grid map by computing elevation statistics. [5] apply a Gaussian process classifier to learn drivable areas using mean elevation features by following a human operator. [8] use two asynchronous threads that compute geometric features and extract semantic information with a lidar. For appearance-based approaches, [9] present a color-based classification system to segment outdoor terrain from camera images using Gaussian Mixture Models (GMMs). [3] apply SURF features together with a random forest classifier to segment outdoor scenes. [10] assess terrain traversability using a bag of visual words model (BOVW) created from SURF features with a support vector machine (SVM) classifier. [11] implement a neural network to extract features from images and apply a fuzzy logic framework to classify rough terrain. These approaches rely on an individual method or use one particular sensor and can only identify terrain traversability in certain types of environments.

However, there are cases where one source of sensory information is not enough. For example, estimating whether a terrain surface would bear the weight of a vehicle or not relies on both geometric and appearance properties of the surface. This motivates methods that combine complementary features from several sensors. [12] present a method for context-sensitive terrain classification based on lidar and camera data. [13] describe an online method to learn the traversability from RGB and stereo cameras between the robot and unknown outdoor environments. [14] and [15] use a near-to-far learning approach which first learns terrain properties in a short range using geometric features and subsequently applies the geometric result to an image-based classifier to perform long range classification. [16] train a deep hierarchical network to extract features from images and apply a self-supervised learning method for long-range vision that can classify complex terrains at distances up to the horizon. The approaches mentioned above require an additional labeling process which takes the result produced by the data of one sensor as label and apply to the data of other sensors. Thus, the data from different sensors need to be processed sequentially. The error from an earlier stage may therefore negatively affect the performance of the next stage. [17] compute two 2D probabilistic traversability maps in parallel from lidar and camera features. We also compute geometric features from a lidar and semantic features from the camera independently. In contrast to [17] we directly fuse the intermediate feature representations of the two sensors instead of fusing the final traversability values. This allows us to fully exploit the joint information of the two complementary classification pipelines.

Most of the approaches above train a classifier from scratch and either apply supervised learning using labeled data or apply self-supervised learning to adapt to the new environment online. To generate reliable results in different environments, these systems require large amounts of training data of the target environment. The data can be hand-labeled by the operator or the robot has to self-label it by driving over different types of terrain. Instead of learning from experience, we train our convolutional network using a dataset with a strong focus on urban semantic assessment. We remove unwanted biases of the urban semantic settings via data augmentation techniques and adapt the network to off-road environments by fine-tuning it with only few hand-labeled images.

III. PROBLEM FORMULATION

Before going into the description of our method we will briefly describe the scenario we target. The context is a search and rescue robot. It has to operate in both urban and off-road settings. This means that it needs to handle scenes that range from mostly urban with cars and buildings to off-road scenarios that involve mainly vegetation and grass. We want the system to be able to operate in a new environment with as little need for new training as possible. The problem is thus to design a method that achieves this goal.

To train and evaluate our system with such different types of environment we have gathered data in six distinct locations around the KTH university campus as seen in Fig. 2. At each location, data is gathered along an approximately 200 meter long trajectory. The data is collected during both summer and winter months to provide maximum variability in the camera images. We collect images from a camera and point clouds from a 3D laser scanner.

We selected eleven equidistant images from each trajectory together with the registered point cloud at the same moment in time. These define the target datasets for training and testing. We hand-labeled the images with pixel-wise annotations into semantic classes and the three traversability categories using LabelMe [18]. A patch of terrain is considered *safe* if the robot can traverse it on its own without encountering problems such as the wheels getting stuck. An *obstacle* patch is reserved for areas that pose a security threat to the robot or the robot to its surroundings, for example vegetation or humans, respectively. A *risky* patch is defined as neither of the other two and typically corresponds to areas where caution is needed.

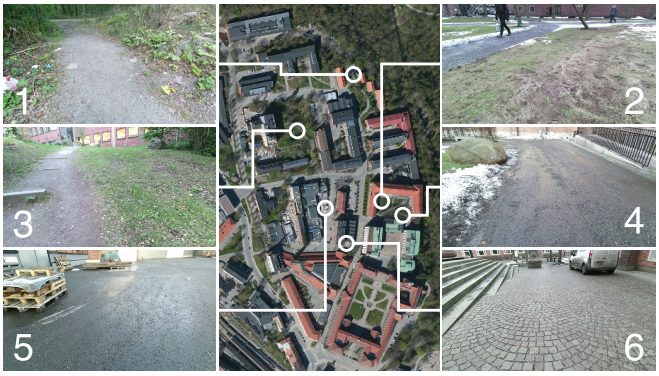


Fig. 2: Point clouds and color images are collected in six different locations around the KTH university campus, representing both urban and off-road conditions. The data was gathered during both summer and winter months to ensure visual variability to test robustness.

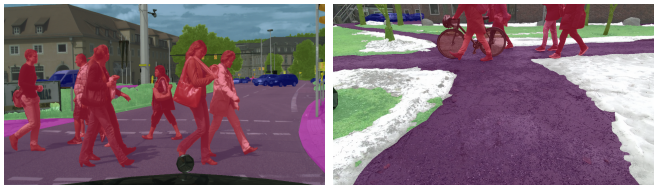
IV. TERRAIN CLASSIFICATION METHOD

Our proposed terrain classification system derives several characteristics from both visual and geometric inputs. The vision-based analysis takes camera images as input and classifies the terrain type into a set of terrain classes. Similarly, the geometry-based analysis takes the terrain elevation point cloud generated from raw laser scanner data as input and provides structural terrain features which can be applied directly to assess the terrain traversability. Sec. IV-A describes the visual terrain segmentation architecture and Sec. IV-B outlines our method for geometric terrain feature extraction. Sec. IV-C describes the fusion of the two subsystems into terrain traversability categories, i.e. an actual cost for a specific robot in a navigation task.

A. Visual terrain features from semantic segmentation

We formulate the vision-based terrain analysis as a pixel-wise semantic segmentation problem. To this end, we employ a fully convolutional network to provide class predictions. The network is trained on a source dataset which already contains several relevant classes for terrain estimation. We use the CityScapes dataset [19] as the source dataset. We believe it provides a strong baseline for visual traversability assessment and navigation. The dataset contains 2975 training and 500 validation images with pixel-wise annotations from a total of 34 semantic classes. It is geared towards autonomous navigation in an urban setting and thus consists of classes such as *car*, *human*, and *traffic light*. The classes which are most relevant for outdoor traversability assessment are *road*, *sidewalk*, *vegetation*, and *terrain*. Fig. 3 provides a side-by-side comparison of typical scenes contained in both source and target environment.

The CityScapes evaluation process assumes that a subset of 15 less relevant classes such as *trailer* and *caravan* are ignored during training. However, our model has the capacity to predict all 34 classes of the original dataset in order to be able to repurpose those which are irrelevant for the environment described in Sec. III.



(a) CityScapes example (b) Campus example

Fig. 3: Sample images with ground truth overlay from the source and target datasets. The main classes involved are *road* (purple), *terrain* (light green), *vegetation* (dark green), *person* (red), *building* (dark gray), and *snow* (white).

1) Training the baseline model: To obtain the baseline semantic segmentation model, we transfer the weights from a VGG-16 [20] network pre-trained on the ImageNet dataset [21] to the corresponding layers of FCN-8s. We initialize the transpose convolution layers to perform bilinear upsampling as described in [22] and the scoring layers with small random weights proportional to the filter size in order to preserve the variance of the activations [23]. We employ dropout with a keep probability of 0.5 in the final convolutionalized layers during training [24]. The model is trained by minimizing the categorical cross-entropy using stochastic gradient descent with momentum of 0.9, a mini-batch size of 10, and a constant learning rate of 10^{-4}. Additionally, we penalize large weights with a weight decay factor of $5 \cdot 10^{-5}$. The images are downsampled by a factor of 4 from the original 1024×2048 to 256×512 pixels. We find that this input size offers the best theoretical performance while simultaneously providing close to real-time inference as can be seen in Tab. I. We additionally standardize the input images across all channels to have zero mean and variance one.

2) Training the augmented model: In order to overcome the urban biases of the CityScapes dataset, we employ aggressive data augmentation techniques which are commonly applied to mitigate problems with overfitting [25]. We downsample the original 1024×2048 images by a factor of 2 and randomly crop 256×512 pixel patches when training and flip the image horizontally at random. Since cropping from the upper part of the image results in crops that mostly contain *building* and *sky* as seen in Fig. 3a, we focus our attention on the lower image regions. Moreover, we randomly adjust brightness, contrast, and saturation by 25% during training.

3) Fine-tuning the model: We employ transfer learning techniques to make the model fully adapt to the target campus environment both in terms of overall classification performance and the incorporation of a new *snow* class as seen in Fig. 3b. To incorporate the new class, we change the semantics of the previously ignored *caravan* class to denote *snow*. We then retrain the model on the 2975 training images of CityScapes and chose a subset of images that contain snow in order to learn the new class. In order to increase the frequency of images from the campus environment during training, we give those images a 30 times higher chance to

TABLE I: FCN-8s inference time and theoretical max. performance with different image downsampling factors

Downsampling factor	1	2	**4**	8	16
Inference time on K40 in s	1.33	0.42	**0.19**	0.14	0.12
Max. mean IU [19] in %	100	97.2	**95.2**	90.7	84.6

be randomly sampled, resulting in a ratio between source and target dataset of roughly 90% to 10%. The data augmentation scheme remains unchanged and we perform gradient descent as before. We fine-tune with a reduced learning rate of 10^{-5} on all trainable parameters.

B. Geometric terrain feature extraction

Our geometry-based terrain estimation algorithm is inspired by the work in [1] and the GESTALT system [2] which is commonly used to assess terrain traverability from point cloud data. In [1] and [2], a 2D traversability grid map is constructed by computing elevation statistics from the set of points within each grid cell. This model represents terrain properties with three quantities: the largest step height, slope and roughness. These features are computed by fitting a plane to points in a terrain patch. The coefficients of the plane are then used to calculate the slope and the plane fitting residuals are used to estimate the roughness. The maximum step height and slope are dictated by the robot model and the roughness represents the unevenness of the terrain patch.

As in [3] and [4], we use the maximum height difference as the maximally traversable step height. We then apply principal component analysis (PCA) to capture the slope and apply the Difference of Normals (DoN) operator [26] to estimate the roughness of each grid cell. The DoN operator computes the difference of normals with multiple radii and shows high point cloud classification performance on the KITTI dataset [27]. Moreover, it is computationally efficient for large scale unorganized 3D point clouds.

C. Combining geometric and visual features

Each of the two terrain traversability estimation methods presented in the previous sections can be run as stand-alone classifiers. However, each method has its own advantages and limitations. The semantic label extracted from the vision-based system contains no geometric information of the terrain. This means that a region in the image that is covered by *grass* or *snow* is likely to be classified as *risky* independent of clearly visible geometric cues. For example, a grass-covered vertical wall that is clearly an obstacle would be classified as only *risky*. The geometry-based features, on the other hand, completely rely on the terrain elevation measured by the laser scanner. This means that when driving on soft or deformable terrain, such as grass or snow, the measured elevation profile may not be the same profile the robot actually experiences.

Considering the limitations of each pipeline, we opt for a fusion method based on concatenating the individual feature vectors and feeding them into a shallow classifier. We compute the geometric features from registered point clouds and project these onto the image plane. For the visual pipeline, we apply the softmax operator to the unscaled per-pixel logits

Fig. 4: Our robot platform is an iRobot ATRV. The robot is equipped with a Velodyne VLP-16 laser scanner and a Microsoft Kinect v2 camera.

of the fully convolutional network to obtain the probability distribution over all 34 semantic classes. For each point in the point cloud which projects onto the image and for which there is a semantic label, the visual and geometric features are then concatenated into a feature vector with a total of 37 individual elements. We use these feature vectors to train a classifier to predict the traversability classes *safe*, *risky*, and *obstacle*.

V. EXPERIMENTAL SETUP

We collected terrain data using a Velodyne VLP-16 lidar and a front-facing Microsoft Kinect v2 camera which are mounted on our four-wheeled robot platform. The lidar has 16 rotating lasers mounted with different pitch angles which can cover the entire $360°$ field of view horizontally and from $-15°$ to $+15°$ vertically. The camera is operating at a 540×960 pixel resolution and is tilted downwards by $15°$ to best cover the terrain in front of the robot. Fig. 4 shows an overview of the vehicle and sensor configuration.

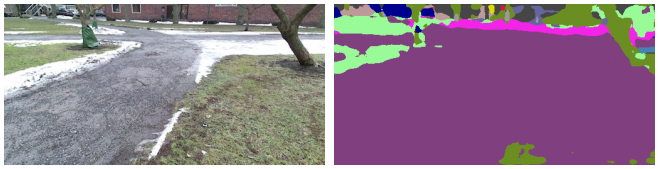
We apply the mapping algorithm presented in [28] to register single Velodyne frames to a dense point cloud. The algorithm updates the point cloud with a frequency of 2 Hz and returns an average of $6 \cdot 10^5$ points for each scan.

VI. EXPERIMENTAL RESULTS

In the following, we present results on both the semantic segmentation performance and our fusion method of visual and geometric features using an off the shelf classifier, in our case a random forest classifier.

A. Semantic segmentation performance

We first evaluate the visual terrain classification performance to investigate the effects of the data augmentation and fine-tuning. We use the standard PASCAL VOC mean intersection over union (IU) metric for semantic segmentation tasks [29]. We compute the mean IU directly from the discretized class predictions of the fully convolutional network. We use bilinear downsampling to resize the original images to a resolution of 256×512 pixels and subsequently standardize them across all channels to have mean of zero and variance of one. No other forms of preprocessing or augmentation are applied during evaluation regardless of the training methodology used.



(a) Input image (b) Baseline segmentation

Fig. 5: Example of the spatial road bias of the *baseline* model. The grass on the bottom right is classified as *road* whereas the grass patches in the upper left are correctly classified as *terrain* albeit having very similar color and texture. Also notice that the model is oblivious to the *snow* present in the scene.

(a) Input image (b) Fine-tuned segmentation

Fig. 6: Example segmentation from the *fine-tuned* model in an unseen location on the KTH campus. The model learns to recognize the *snow* present in the scene, along with all other relevant classes from the CityScapes dataset.

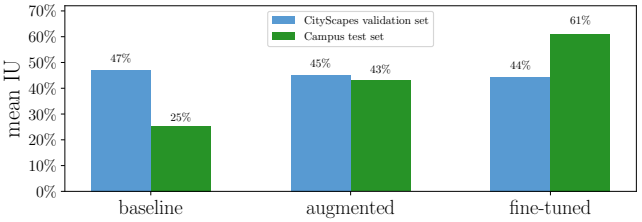


Fig. 7: Mean intersection over union (mean IU) performance of the *baseline*, *augmented*, and *fine-tuned* models evaluated on the CityScapes validation set and the campus test set.

The *baseline*, *augmented*, and *fine-tuned* models are trained as described in Sec. IV. We evaluate their predictive performance on both the CityScapes validation set and on hand-labeled images from the target campus environment. For CityScapes, the mean IU is computed for all 500 validation images and all relevant classes as described in [19] are taken into account. For the target environment, we compute the mean IU on 22 hand-labeled images, not used for training, from two distinct locations and seasons according to the CityScapes labeling policy. However, we make one slight modification and regard all *road*, *sidewalk*, and *ground* predictions as *asphalt*. We find this change in semantics to be necessary for unambigous and unbiased ground truth labels since much of the campus is made of asphalt that has no apparent *road* or *sidewalk* character.

The *baseline* model has the weakest performance in the target environment with a mean IU score of 25.34%. The network has a tendency to overfit the spatial relationships of urban scenes from the perspective of a car operating in traffic. Specifically, there is a large bias towards classifying the central lower region of the scene as *road* regardless of the actual ground material as seen in Fig. 5. Another example is the proneness to identify *cars* towards the border regions of the field of view which likely stems from the ubiquity of parked vehicles on the side of the road in the source dataset. Generally, being able to infer semantics based on spatial relationships in the scene is a desirable property for segmentation algorithms. However, assuming that the vehicle is driving on a road at all times is too strong of a bias for the target campus environment.

The *augmented* model can alleviate many of these spatial and relational biases and achieves a mean IU of 43.00%. We attribute the boost in performance to the aggressive data augmentation scheme employed during training. By sampling patches from different regions of the image we can weaken the spatial biases and achieve more robust segmentations in our target environment. Moreover, we can achieve a comparable performance on the target environment and the CityScapes dataset, on which we obtain a mean IU of 45.22%.

The *fine-tuned* model exhibits the best performance on previously unseen locations in the target environment with a mean IU score of 60.94%. The model quickly learned to recognize the new class *snow* and to adapt to the campus environment as seen in Fig. 6.

An overview of the experimental results on both the CityScapes evaluation set and the campus test set can be obtained in Fig. 7. The *augmented* model adapts to the spatial structure of new environment through data augmentation and the *fine-tuned* model quickly learns to recognize *snow* by fine-tuning on a small set of eleven annotated images. On CityScapes, we can observe a slight but noticeable performance drop of the extended models in comparison to the baseline. We attribute the drop from *baseline* to *augmented* and *fine-tuned* model to the fact that we trained on crops but still evaluated on the entire field of view and the addition of the new class, respectively.

During our live experiments on the robot platform, we noticed severe perceptive failures of the visual pipeline when traversing *risky* surfaces such as *terrain* and *snow*. The most likely explanation is that the adverse effects are caused by motion blur. We simulate artificial motion blur in our target environment by applying filters of different sizes to the color images. We then re-compute the mean IU for the entire test set. An example for a 3×3 motion-blurring filter is given by

$$F = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix}$$

and applied to each image channel individually. Larger filters are created in the same fashion with $1/s$ on the center horizontal for a corresponding $s \times s$ filter where s is odd.

Fig. 8 provides an example of how the different kernel

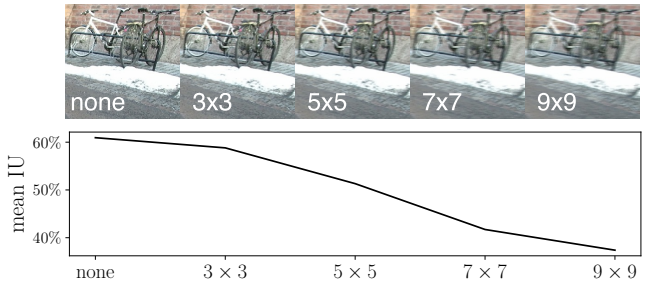


Fig. 8: Mean intersection over union (mean IU) performance of the *fine-tuned* model evaluated on the campus test set with different levels of horizontal motion blur.

sizes affect the input images and relates these to the resulting mean IU performance over the whole dataset. Our *fine-tuned* model is not able to cope with the motion blur although the scene remains perfectly recognizable for a human observer. However, we can identify the adverse effect that motion blur has on the predictive power of convolutional networks. This is in line the results in [30] whose experiments with Gaussian blur show similar detrimental effects on classification performance of convolutional networks.

B. Evaluation of fusing semantics and geometry

We select two labeled subsets from urban environments which contains rich geometric and semantic information as training sets and the four remaining subsets as test sets. We evaluated three different classifiers: decision tree, random forest and support vector machine. The following experimental results are based on a random forest classifier which shows a good trade-off between accuracy and training efficiency.

We present quantitative results for the performance of our algorithm in urban, off-road, and mixed environments. Fig. 10 shows the receiver-operating-characteristic (ROC) curves for classification results of each terrain class using geometry-based features, vision-based feature and the combined feature vector. We can observe that vision performs better than the geometric classifier on detecting *safe* terrain but performs worse on detecting the *obstacles*. This can be explained since the elevation of *safe* and *risky* terrain captured by the lidar is too similar such that the visual features dominate the traversability class decision. The same is true for *obstacle* detection where geometric features have more impact than the semantic features.

Risky terrain is challenging for both individual classifiers since the traversability class depends strongly on the interplay between visual and geometric features. An example of classifying off-road terrain is shown in Fig. 9. We can see that the vision classifier assesses the rock as *risky* instead of *obstacle* as there is grass on top of it. The geometric classifier is not able to separate the road from grass as the two terrains share similar geometric properties. The classification performance improves significantly when we train the random forest on combined features.

VII. CONCLUSIONS

We present a multi-sensor terrain classification system based on geometric and semantic features which can be applied in urban, off-road, and mixed environments. Unlike other online approaches that learn from experience only, we apply transfer learning using a dataset which provides a strong baseline for urban semantic segmentation. By performing data augmentation and fine-tuning, we remove the biases of the urban environment and allow the model to robustly classify terrain in off-road environments, including scenes that contain *snow*. We then combine the semantic labels with geometric features to present a more generalized description of the terrain traversability. It allows the system to robustly classify terrain in novel environments without retraining. Finally, we apply a random forest classifier to categorize the terrain traversability into three classes: *safe*, *risky* and *obstacle*. We implement the algorithm on our four-wheeled robot and provide experimental results on datasets which are collected at different locations. The results show that our system can perform accurate classification of terrain traversability in novel mixed environments.

In future work, we plan to combine our algorithm with an online learning approach that can adapt to new environments by exploring them systematically. Moreover, we plan to investigate how the adverse effects of motion blur on convolutional neural networks can be mitigated. In future work we also want to explore replacing the three geometric features with a convolutional neural network model based on disparity maps generated from the point cloud. Instead of using simple classifiers such as random forest or support vector machines, one could apply a high level feature fusion strategy that can learn the interaction between each sensor in order to produce more reliable results.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 644839 (CENTAURO). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU used for this research.

REFERENCES

[1] D. B. Gennery, "Traversability Analysis and Path Planning for a Planetary Rover," *Autonomous Robots*, vol. 6, no. 2, 1999.

[2] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo Vision and Rover Navigation Software for Planetary Exploration," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 5. IEEE, 2002.

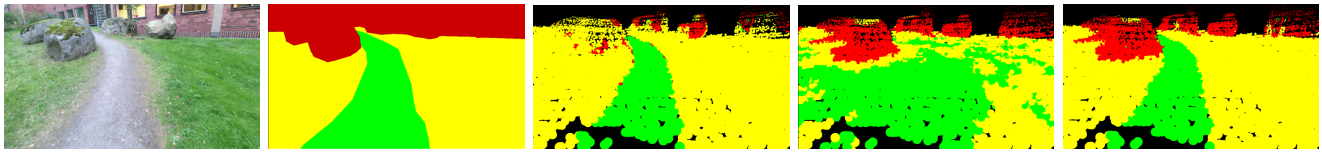
[3] Y. N. Khan, P. Komma, and A. Zell, "High Resolution Visual Terrain Classification for Outdoor Robots," in *ICCV*. IEEE, 2011.

[4] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain Characterization and Classification with a Mobile Robot," *Journal of Field Robotics*, vol. 23, no. 2, 2006.

[5] L.-P. Berczi, I. Posner, and T. D. Barfoot, "Learning to Assess Terrain from Human Demonstration using an Introspective Gaussian-Process Classifier," in *ICRA*. IEEE, 2015.

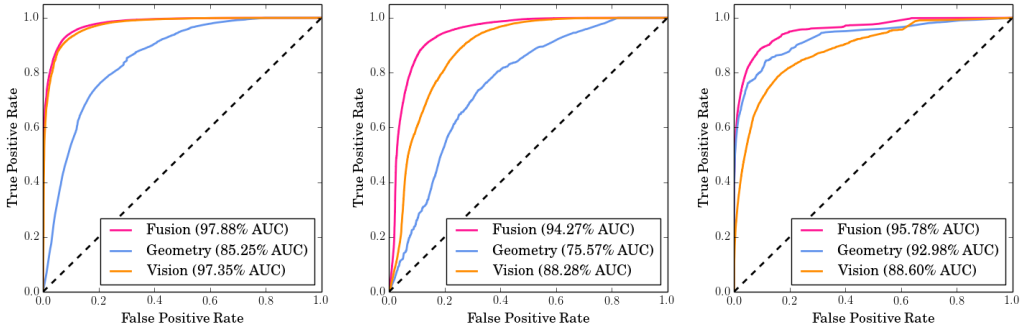
[6] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, 2013.

[7] J. Schmidt and A. Hewitt, "Fuzzy land element classification from DTMs based on geometry and terrain position," *Geoderma*, vol. 121, no. 3, 2004.



(a) Input image (b) Ground truth (c) Vision only (d) Geometry only (e) Fusion result

Fig. 9: Classification results using a random forest trained on visual features only, geometric features only, and the combined feature vector. The visual features are not able to reliably classify the rock as an *obstacle*, whereas the geometric features cannot reliably classify the grass as *risky*. The fused result offers the best assessment from the two complementary pipelines. Green denotes *safe*, yellow denotes *risky*, and red denotes *obstacle*.



(a) Safe (b) Risky (c) Obstacle

Fig. 10: ROC curves for the classification result of three terrain classes using a random forest trained on visual features only, geometric features only, and the combined feature vector. The visual features can reliably classify *safe* terrain while the geometric features are stronger in classifying *obstacle*s. The combined feature vector provides the most accurate classification results.

[8] B. Suger, B. Steder, and W. Burgard, "Terrain-Adaptive Obstacle Detection," in *IROS*. IEEE, 2016.

[9] P. Jansen, W. van der Mark, J. C. van den Heuvel, and F. C. Groen, *Colour based Off-Road Environment and Terrain Type Classification*. Piscataway, NJ: IEEE, 2005.

[10] P. Filitchkin and K. Byl, "Feature-Based Terrain Classification for LittleDog," in *IROS*. IEEE, 2012.

[11] A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *Journal of Field Robotics*, vol. 18, no. 10, 2001.

[12] S. Laible, Y. N. Khan, and A. Zell, "Terrain classification with conditional random fields on fused 3d lidar and camera data," in *ECMR*. IEEE, 2013.

[13] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. F. Bobick, "Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation," in *ICRA*. IEEE, 2006.

[14] M. Bajracharya, A. Howard, L. H. 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, 2009.

[15] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised Monocular Road Detection in Desert Terrain," in *Robotics: science and Systems*, vol. 38. Philadelphia, 2006.

[16] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning Long-Range Vision for Autonomous Off-Road Driving," *Journal of Field Robotics*, vol. 26, no. 2, 2009.

[17] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic Traversability Map Generation using 3D-LIDAR and Camera," in *ICRA*. IEEE, 2016.

[18] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *IJCV*, vol. 77, no. 1-3, 2008.

[19] 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 *CVPR*, 2016.

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, 2015.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, June 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012.

[26] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds," in *3DIMPVT*. IEEE, 2012.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*. IEEE, 2012.

[28] D. Droeschel, J. Stückler, and S. Behnke, "Local Multiresolution Representation for 6D Motion Estimation and Mapping with a Continuously Rotating 3D Laser Scanner," in *ICRA*. IEEE, 2014.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *IJCV*, vol. 88, no. 2, jun 2010.

[30] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks," 2016. [Online]. Available: http://arxiv.org/abs/1604.04004