Online Learning Terrain Classification for Adaptive Velocity Control

Wei Mou
University of Freiburg
Georges-Köhler-Allee 52
79110 Freiburg, Germany
mouwei@informatik.uni-freiburg.de

Alexander Kleiner
University of Freiburg
Georges-Köhler-Allee 52
79110 Freiburg, Germany
kleiner@informatik.uni-freiburg.de

Abstract — Safe teleoperation during critical missions, such as urban search and rescue and bomb disposal, requires careful velocity control when different types of terrain are found in the scenario. This can particularly be challenging when mission time is limited and the operator's field of view affected.

This paper presents a method for online adapting robot velocities according to the terrain classification from vision and laser readings. The classifier adapts itself to illumination variations, and can be improved online given feedback from the operator.

Keywords: *Terrain Classification, HRI, Self-supervised Learning, SVM*

I. Introduction

Teleoperation during critical missions, such as urban search and rescue and bomb disposal, requires careful velocity control when overcoming different types of terrain. However, operators are typically working under high pressure and are confronted with a limited field of view making the safe navigation of the robot a challenging task. One solution to this problem is to support teleoperation with automatic velocity control based on the classification of the terrain.

A variety of approaches based on different sensors for terrain classification tasks have been introduced in the past. Most existing work focuses on segmenting a traversable surface from a geometric or non-geometric hazard rather than specifying the terrain types. For laser based approaches as described in [1], geometric obstacles like trunks or rocks are separated from traversable area by using a rule-based classifier. Traversable terrain can be detected by using vision-based approaches utilizing color or texture features, as proposed in [2] and [3]. In [4] and [5] authors proposed terrain classification methods based on the sensing of vibration. The visual information of wheel sinkage is introduced in [6] to classify terrains under the robot. Despite the efficiency and good experimental results in [5] and [6], the wheels have to have contact with the surface in order to make vibration measurements, which can be in some cases too late for speed adjustments. To overcome the problems of individual sensors, researchers have combined different sensors for better classification results. In [7] prediction results of the same terrain patches from laser and vision are combined for better performance than the single methods applied on their own. In [8] authors uses classification results from laser sensors as input for the self-supervised learning of the vision model to find the road. Although this approach has proven to be successful and robust to terrain and illumination changes during the Darpa Grand Challenge 2005, their proposal does not specify terrain types and is mainly tailored for desert terrain. Vibration and vision sensing are combined in [9] and [10]. The system in [10] provides a more accurate prediction than using vibration measurements alone. However, it lacks robustness against illumination variation since the trained SVM cannot be adjusted online and it only classifies surfaces the robot has already traversed. In [9] visual appearance is learned based on vibration sensing and is used to identify the distant terrain and the vision models can be updated on-line. However, the adaption of this approach is not as efficient as in [8], and only three different terrain types are classified. Because the on-line learning is achieved by repeatedly training a support vector machine (SVM) whose training set size is limited to 400 samples and the training procedure for SVM is time consuming, the system cannot adapted to changes in real-time. To overcome the limitations of most existing approaches, and to achieve the level of accuracy and fast adaption in real world tasks, a self-supervised learning

Fig. 1: Terrain types that are detected by the presented system: 1. Grass. 2. Asphalt. 3. Gravel. 4. Pavement. 5. Indoor Floor.

approach based on multiple sensors is presented in this paper. The self-supervised learning is achieved by combining vision and vibration sensing. Local vibration sensors are applied to identify terrain classes. A visual model is trained based on the color information extracted from the ground patch the robot traverses. The vision model classifies the relevant pixels in the images, and provides prediction results on the surface in front of the robot. A normalization method is applied for the vision classifier to reduce the variance from illumination changes. In some cases, different terrains may have a similar appearance that is difficult for the vision classifier to separate. To overcome this limitation and to make the system more robust to lighting conditions, a laser-based classifier is utilized additionally. By fusing the results from vision and laser classifiers, the accuracy of the classification is improved as compared to using each classifier individually. Experiments were carried out to demonstrate the system's capability by distinguishing five different types of terrain (Grass, Asphalt, Gravel, Pavement, Indoor Floor), as shown in Figure 1.

The presented results show that the system is able to perform online self-supervised learning and to predict the terrain in front of it. Consequently, the system determines the appropriate speed for different types of terrain. To deal with misclassification, human-robot interaction is applied to adjust the system's velocity settings by providing feedback via a joypad. This adjustment is then learned by the system to update its internal model.

The remainder of the paper is structured as follows: The vibration-based classifier, the vision-based classifier, and the laser-based classifier are described in Section II, Section III, and Section IV, respectively. Section V describes the approach of combining these classifiers. Section VI presents how a human operator provides feedback for updating the classifiers, and improving the classification results. The performance of our proposals is demonstrated in Section VII. Conclusions and future works are discussed in Section VIII.

II. Vibration-based classifier

In this section a supervised vibration-based classifier using a support vector machine (SVM) is presented. It is inspired from the method introduced in [5]. Vibrations on the robot can be measured by accelerations along three perpendicular directions: front-back (X-Axes), left-right (Y-Axes), and up-down (Z-Axes). The measured data is segmented into groups with a size of 100 samples, and features are extracted from each group. For each axes, 8 features are defined (as in [5]) extracted and normalized to the interval [-1, +1]. Unlike [5], in which the robot is steered with a constant velocity, our system adapts velocities to ensure the safety of the robot. Additionally a feature V_t is added into feature vectors where V_t is the average translation velocity for each group's measurements. So each feature vector contains 25 features. These features are labeled by hand and an one-against-all multi-class support vector machine (SVM) is learned offline. SVM is a machine learning technique that constructs a hyperplane to separate two classes and maximize the margin between the closest point

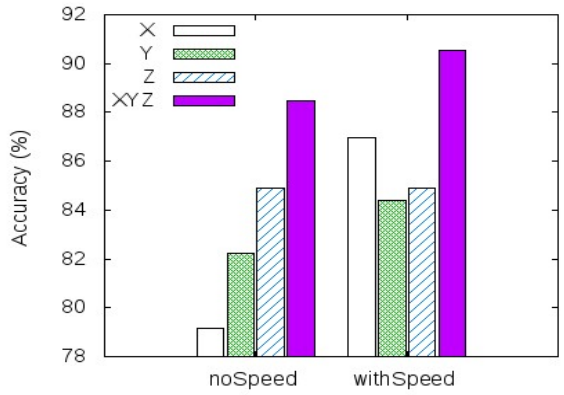


Fig. 2: 10-Fold Cross Validation Results detecting the state of the robot from measured accelerations. X (front-back), Y (left-right), Z (up-down) illustrate the detection accuracy on different axes.

from each class to it. In the case of non-separable data, a soft margin is used which allows for a small training error. The training problem becomes a trade-off between a large margin and a small error penalty. The penalty of an error is assigned by a parameter C which is chosen by the user. A larger C corresponds to a higher penalty to errors. A non-linear hyperplane is constructed by using kernel functions to map data into a higher dimensional space in which data can be separated linearly. In our case a Radial Basis Function $K(x_1, x_2) = exp(-\|x_1 - x_2\|^2/2\sigma^2)$ is used as kernel function and the parameters of error penalty C and RBF width σ are tuned by grid search. The trained SVM is tested by 10-fold cross validation and the result is shown in Figure 2.

For the prediction the acceleration data is collected by the robot and feature vectors are extracted and normalized in the same way as in the training phase. Then the SVM is applied to online classify each feature vector into a corresponding terrain type. The SVM classifier is implemented by using the LIBSVM [11]. The result for this classifier is used to label the training data for the vision-based classifier presented in the next section.

III. Vision-based classifier

(a) Original Image (b) Normalized Image

Fig. 3: Image normalization and sky detection: The red line in (b) indicates the detected horizon line. The pixels above this line are all considered as irrelevant points to our task.

We used a forward looking monocular camera mounted on the robot for image capture. In each captured image, pixels from sky and shadows are irrelevant to our task. Hence, to save computation time, these pixels are removed beforehand by using a horizon finding algorithm as introduced in [12]. In Figure 3(b), the pixels above the horizon line are considered as sky and are eliminated.

In order to reduce the illumination variation and increase the contrast of the image, the brightness distribution of an image is equalized. Because the color balance should be kept during the equalization process, we first convert the image from RGB space to YCbCr space and then apply equalization only on the Y channel. Figure 3 shows the equalization result.

For associating image pixels with the current state of the robot, the robot's pose is projected onto images from the past. The current position of the robot in previous images can be estimated by the projection computed from the camera parameters resulting from calibration [13] and the 3D pose of the robot. By this, image patches that are traversed by the robot are extracted and marked as training set. These areas are then labeled by the result from the vibration-based classifier described in the previous section.

After the training area is extracted from the image, a color based model is trained to predict the types of terrain in front of the robot. For training and prediction we improve the approach proposed in [8] in which Gaussian Mixture Models (GMMs) are applied to extract the traversable surface from the images. Robustness is achieved by combining laser and vision to proceed self-supervised learning. In contrary to the method described in [8] we use multiple GMMs to model the terrain type in front of the robot.

Each type of terrain is modeled by a GMM in RGB space. The five different types of terrains are represented by five GMMs. Given the training area of the image and its label, each GMM can be trained by a self-supervised learning approach proposed in [8]. The K-Means clustering method is used to update each Gaussian in the GMM, which makes the system more robust to lighting and color changes. Each Gaussian is modeled by its average value, its covariance matrix, and its mass (number of pixels in the Gaussian).

After the learning step, each terrain type is represented by a GMM containing several Gaussians. We can score each related pixel in the image by using the Mahalanobis distance between the pixel and each GMM. The distance for each GMM is represented by the minimal distance between the pixel and each gaussian of the GMM. The matching score for pixel p_i and one gaussian m_j in each GMM is defined as

$$s(p_i, m_j) = \frac{1}{dist(p_i, m_j)} * mass(m_j)$$

where $dist(p_i, m_j)$ is the Mahalanobis distance from pixel p_i to Gaussian m_j and $mass(m_j)$ is the normalized mass for Gaussian j and $mass(m_j) \in [0, 1]$. The matching score of pixel p_i and a particular GMM M_k is defined as

$$S(p_i, M_k) = \max_{j \in [1,n]} s(p_i, m_j)$$

where n is the number of Gaussians in M_k. The classification of a pixel in the image is found by:

$$\arg\max_k S(p_i, M_k), k \in [1, m]$$

where m is the number of GMMs and class label k is assigned to pixel p_i. The classification result is shown in Figure 4. The left image is the latest frame taken from the camera. The middle one is the classification result for the vision classifier. The green region represents the grass and the blue one represents the road. The right side is the picture from a previous frame and the red trapezium indicates the current position of the robot. The pixels inside the trapezium are extracted as training data for the vision classifier and are labeled by the current classification result of the vibration classifier.

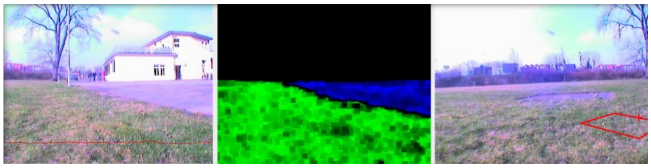


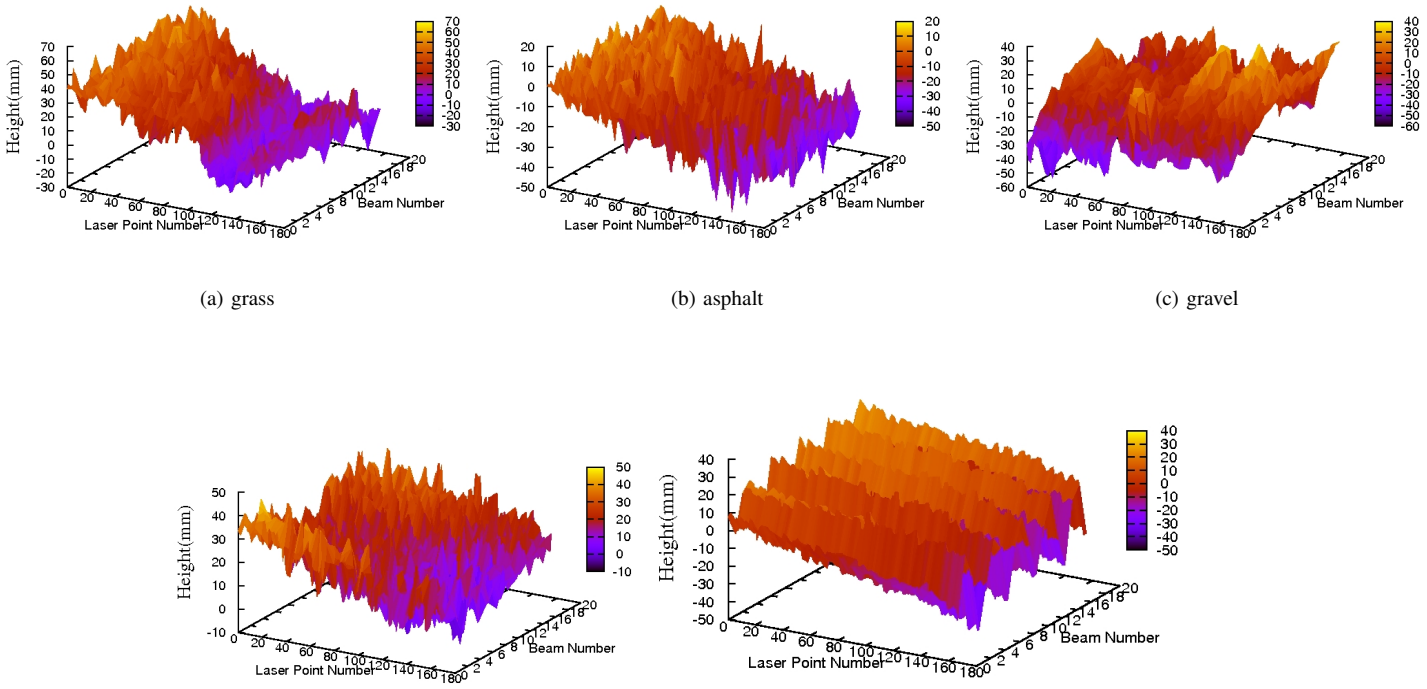
Fig. 4: Vision-based classification result for grass and asphalt.

IV. Laser-based Classifier

In an urban environment, specific types of terrains can have very similar color appearance, as shown in Figure 1. It is difficult for a color based classifier to separate these terrains. When the robot drives onto a new type of terrain, the difference is detected by the vibration based classifier triggering an update of the vision-based model. However, this direct update can be too late and thus the unchanged velocity could cause to damage the robot. For example, the vision model could misclassify the gravel as asphalt because of their color similarity. As a result, the robot would drive on to gravel in a relatively high speed. To cope with this failure and to increase the classification accuracy, we incorporate a laser-based classifier into the system.

The laser scan lines are accumulated over time and only the height information of these laser points is used. The height data is split into small segments where each segment represents the robot's translation in half a second. Features are extracted from these segments and later used for learning and prediction. For each height segmentation, we define a two dimensional matrix $H = h_{ij}, i \in [1, N]$ and $j \in [1, M]$ where N is the number of laser points in a single laser scan line and M is the number of scan lines in the segmentation. Also we define a vector V which is converted from all h_{ij} into a one dimensional vector. The features are extracted as the following:

1) The average height value of H. $\mu = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{h_{ij}}{mn}$.
2) The maximum value for all h_{ij}.
3) The minimum value for all h_{ij}.



(a) grass (b) asphalt (c) gravel

(d) pavement (e) indoor

Fig. 5: Laser shapes of different terrain. A tracked robot is used so that even on flat surfaces, such as indoor floor, the LRF can shake significantly as shown by (e).

4) The standard deviation of V. The coarser a surface, the higher value of the standard deviation is.
5) The sum of the square value of H, $sum = \sum_{i=1}^{n}\sum_{j=1}^{m} {h_{ij}}^2$. For a coarse surface this value is higher than a smooth surface even if the average heights are similar.
6) The sum of the standard deviation $\sum_{1}^{n}\sigma_i$ where σ_i denotes the standard deviation of the height data from the i^{th} laser scan line.
7) $\sum_{1}^{m}\sigma_j$ where σ_j denotes the standard deviation of the height data from the j^{th} point of all laser scan lines.
8) The number of times that data in V traverses over the mean of V. This provides the main frequency of the signal.

Those 8 features extracted from gathered raw data are labeled by hand and used for training. Like the vibration based classifier the data is normalized and trained with a SVM. After the training phase, the 10-fold cross validation accuracy of the classifier reached 84.71%. Finally, we use this SVM for prediction. The same features are extracted from raw laser data as input to the SVM for predicting the terrain types in front of the robot.

V. Combination of Vision-based and Laser-based Classifier

As shown in Figure 5, the results (shapes) of laser scan for grass and asphalt are quite similar whereas the results (colors) of the vision-based classifier are significantly different. Hence, a combination of both classifiers can produce a more reliable prediction result.

Here a naive Bayes classifier is applied to fuse both classifiers. It combines the naive Bayes probability model with a decision rule. One common rule is to pick the hypothesis that is most likely, which is known as the maximum a posteriori or MAP decision rule. The classifier is defined as follows:

$$classify(v,l) = \arg\max_{c_i \in C} P(c_i \mid v,l) \tag{1}$$

where C is the set of terrain types. v and l are the classification results from the vision and laser based classifier, respectively. With Bayes rule applied to $P(c_i \mid v,l)$, the following equation is derived:

$$P(c_i \mid v,l) = \frac{P(v,l \mid c_i)P(c_i)}{P(v,l)} \tag{2}$$

Vision and laser classifiers are two approaches using different sensors and different models. As a result, assume that v is

conditionally independent from l. With this assumption, we formulate:

$$P(v, l \mid c) = P(v \mid c)P(l \mid c)$$

and

$$P(v, l) = P(v)P(l)$$

Hence, equation (1) can be written as:

$$\arg\max_{c_i \in C} P(c_i \mid v, l) = \arg\max_{c_i \in C} \frac{P(v \mid c_i)P(l \mid c_i)P(c_i)}{P(v)P(l)} \quad (3)$$

$$= \arg\max_{c_i \in C} P(v \mid c_i)P(l \mid c_i)P(c_i) \quad (4)$$

Note that we can drop $P(v)P(l)$ since this term is constant and independent from the hypothesis. The likelihood and prior can be estimated based on the frequencies in the training data. Each group of data consists of the triple: G(ground truth), R_v (prediction result from the vision classifier) and R_l (prediction result from the laser classifier).

$$P(v \mid c_i) = \frac{\phi_{v,c_i}}{\phi_{c_i}} \quad (5)$$

$$P(l \mid c_i) = \frac{\phi_{l,c_i}}{\phi_{c_i}} \quad (6)$$

$$P(c_i) = \frac{\phi_{c_i}}{\sum_{i=1}^{n} \phi_{c_i}} \quad (7)$$

where ϕ_{v,c_i} denotes the number of training samples whose ground truth is c_i and vision label is v. ϕ_{c_i} denotes the number of training samples whose ground truth is c_i. In (7) n is the total number of terrain types. We could also assume a uniform distribution for the prior $P(c_i)$ which means the probability of the robot traversing on any kind of terrain is the same. Given the label of terrain in front of the robot, the driving velocity that ensures safe and fast driving can be decided. The advantage of the naive Bayes classifier is that it can provide the probability distribution of all hypotheses, making it simple to update the classifier online by changing the probabilities of prior and likelihood. This mechanism allows it to a human operator to update the classifiers when a misclassification occurs.

VI. Human Interaction

The classification results can assist human teleoperation by changing the driving style automatically. Whenever misclassification is observed by the operator, feedback can be provided for updating the classifier. When the robot drives with a wrong velocity due to misclassification of the terrain, the operator can use the joystick to overwrite the autonomous velocity control for enforcing the appropriate setting. The robot will update both the vision classifier and Bayes classifier based on the feedback. Online updating the vibration and laser-based classifiers is omitted since they are independent from environmental changes. The model update takes place by the following steps:

TABLE I: Confusion Matrices for 4 Classifiers

	Actual Terrain Label				
Vibration Classifier	Grass	Asphalt	Gravel	Pavement	Indoor
Grass	98.31%	0.35%	0.0%	0.0%	0.0%
Asphalt	0.42%	77.38%	0.37%	0.5%	6.40%
Gravel	0.0%	1.93%	93.66%	11.0%	0.67%
Pavement	0.42%	3.10%	5.97%	88.50%	1.35%
Indoor	0.85%	17.24%	0.0%	0.0%	91.58%

	Actual Terrain Label				
Vision Classifier	Grass	Asphalt	Gravel	Pavement	Indoor
Grass	97.42%	3.99%	0.0%	0.0%	0.0%
Asphalt	0.95%	82.02%	5.16%	27.09%	8.82%
Gravel	0.95%	1.24%	89.25%	2.03%	0.0%
Pavement	0.19%	5.03%	5.59%	70.88%	0.0%
Indoor	0.49%	7.72%	0.0%	0.0%	91.18%

	Actual Terrain Label				
Laser Classifier	Grass	Asphalt	Gravel	Pavement	Indoor
Grass	80.47%	2.10%	8.43%	0.0%	0.0%
Asphalt	4.65%	90.51%	1.4%	13.52%	2.23%
Gravel	10.93%	0.98%	89.12%	0.47%	1.21%
Pavement	3.95%	4.98%	1.05%	86.01%	0.81%
Indoor	0.0%	1.43%	0.0%	0.0%	95.75%

	Actual Terrain Label				
Bayes Classifier	Grass	Asphalt	Gravel	Pavement	Indoor
Grass	97.16%	0.0%	1.76%	0.0%	0.73%
Asphalt	0.53%	95.75%	0.2%	4.26%	1.71%
Gravel	1.78%	0.63%	98.04%	2.20%	0.0%
Pavement	0.18%	3.62%	0.0%	93.54%	0.0%
Indoor	0.35%	0.0%	0.0%	0.0%	97.56%

1) The operator passes feedback to the robot by overwriting the automatically selected velocities with the joypad. The velocity and terrain relationships are pre-defined. So each terrain type has a corresponding velocity. When the robot misclassifies the terrain, it determines the correct label by selecting the terrain with the highest probability that has a higher or slower velocity associated than the current velocity.
2) The vision model is updated by replacing the Gaussians with minimum confidence in the GMM_L by newly generated Gaussians extracted from the current clustering of the image. Unlike the training phase for the vision classifier, in which the training data is labeled by the prediction result of the vibration classifier, the data labeled by the human is more reliable. Consequently, higher confidence values are assigned to the newly added Gaussians.
3) Given the correct label L, the Bayes classifier is updated by changing the prior and likelihood calculated in (5), (6), and (7). Given the current vision class label v_i and laser class label l_j, the update can be accomplished by adding new triples (L, v_i, l_j) for training the Bayes Classifier. The triples are added until the prediction result of the Bayes classifier is L.

The updated models are used to classify the terrain in front of the robot and the corresponding velocity is set according to the classification result.

VII. Experiment Results

Experiments were performed using the Matilda platform from Mesa Robotics (see Figure 6) driven at a maximal

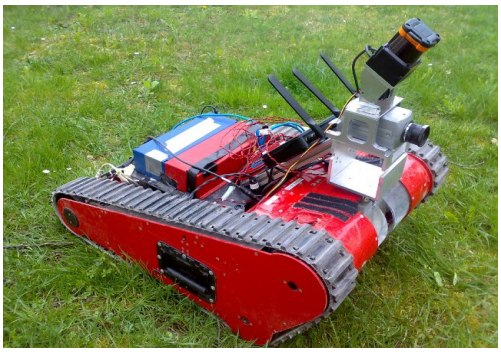


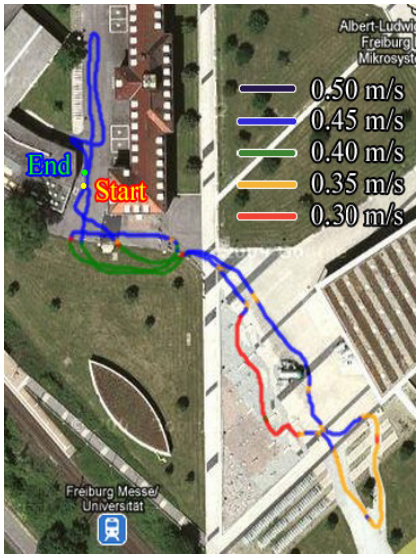
Fig. 6: Matilda Robot with a calibrated camera and a tilted LRF mounted on its head. The Xsens is mounted inside the Matilda which cannot be seen in this picture.

translation velocity of $0.5\ m/s$, The robot was equipped with a Xsens MTi sensor (100 Hz) to measure the 3D acceleration, a monocular forward looking camera, and a 2D Hokuyo UTM30 laser (40 Hz) which is pointed forward along the driving direction inclined as shown in Figure 6. The robot was manually controlled with a wireless joypad. Driving velocities are chosen automatically based on the terrain prediction. The training data has been gathered by steering the robot manually through the environment. The translational and rotational velocities were automatically selected by the presented method. The vibration and laser classifier has been trained with 1014 acceleration segmentations and 1472 laser segmentations. The bayes classifier was trained on 1779 samples. The experiment has been conducted by driving the robot over all five different terrain types for about 30 minutes. The confusion matrices of the results from the three raw data classifiers and the Bayes classifier are shown in Table I. As can be seen from Table I, that the laser model misclassifies grass and asphalt, but the significant color difference makes it easy for the vision model to separate them. The combined Bayes classifier provides better result than using each single classifier. The driving velocities during the test phase are indicated in Figure 7.

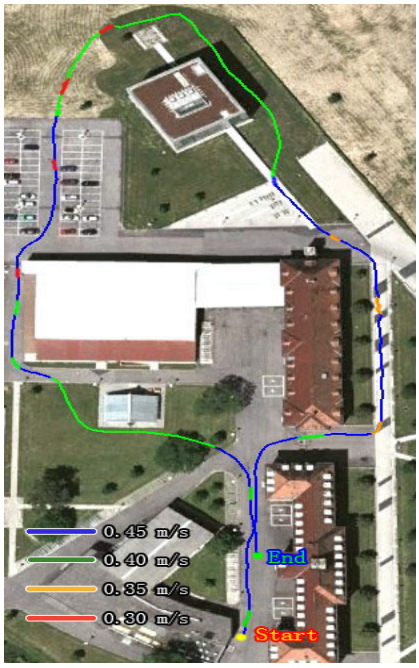
VIII. Conclusion

In this paper we presented a system that can assist human operators to drive the robot in an urban environment. The system can adjust the driving velocity to keep the balance between fast navigation and the safety of the robot. We implement the approach with self-supervised learning which can classify 5 different types of terrain in an urban environment. Multiple sensors are used and 4 classifiers are built and combined to provide the final classification result, which is robust towards changing illumination and more accurate than using any single classifier alone.

There are some points left for future research. Five different types of terrain are defined in this paper. The system can be significantly improved by detecting more types of hazards found in outdoor terrain. For example, the detection of curbs or debris that can slowly be overcome by tracked platforms.



(a) Experiment 1



(b) Experiment 2

Fig. 7: Automatic velocity adjustments during the experiment. Each color on the trajectory denotes a different velocity setting chosen by the system.

Furthermore, the texture character of terrains can be considered for the vision based classifier.

References

[1] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and reliable obstacle detection and segmentation for cross-country navigation," in *In IEEE Intelligent Vehicles Symposium*, 2002, pp. 610–618.

[2] H. Zhang and J. Osuowski, "Visual motion planning for mobile robots," *Transactions on Robotics and Automotion*, vol. 18, pp. 199–208, 2002.

[3] J. Fernandez and A. Casals, "Autonomous navigation in ill-structured outdoor environments," in *in Proc. Int. Conf. Intelligent Robots and Systems*, 1997.

[4] K. I. C. Brooks, "Vibration-based terrain classification for planetary exploration rovers," in *In IEEE Transactions on Robotics*, vol. 21, no. 6, December 2005, pp. 1185–1191.

[5] C. Weiss, H. Frhlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 4429–4434.

[6] S. D. K. Iagnemma, C. Brooks, "Visual, tactile, and vibration-based terrain analysis for planetary rovers," in *In IEEE Aerospace Conference*, 2004.

[7] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *In IEEE International Conference on Robotics and Automation*, 2002, pp. 4320–4325.

[8] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[9] K. I. C. Brooks, "Self-supervised classification for planetary rover terrain sensing," December 2007, pp. 1 – 9.

[10] H. T. C. Weiss and A. Zell, "A combination of vision- and vibration-based terrain classification," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2008, pp. 22–26.

[11] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[12] M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," in *IEEE/RSJ Int Conf on Robots and Systems*, 2002, pp. 2134–2140.

[13] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *in ICCV*, 1999, pp. 666–673.