An Adaptive Outdoor Terrain Classification Methodology using Monocular Camera

Chetan J., K Madhava Krishna, C. V. Jawahar
International Institute of Information Technology, Hyderabad, India-500032.

***Abstract*— An adaptive partition based Random Forests classifier for outdoor terrain classification is presented in this paper. The classifier is a combination of two underlying classifiers. One of which is a random forest learnt over bootstrapped or offline dataset, the second is another random forest that adapts to changes on the fly. Posterior probabilities of both the static and changing/online classifiers are fused to assign the eventual label for the online image data. The online classifier learns at frequent intervals of time through a sparse and stable set of tracked patches, which makes it lightweight and real-time friendly. The learning which is actuated at frequent intervals during the sojourn significantly improves the performance of the classifier vis-a-vis a scheme that only uses the classifier learnt offline or at bootstrap. The method is well suited and finds immediate applications for outdoor autonomous driving where the classifier needs to be updated frequently based on what shows up recently on the terrain and without largely deviating from those learnt at bootstrapping. The role of the partition based classifier to enhance the performance of a regular multi class classifier such as random forests and multi class SVMs is also summarized in this paper.**

I. Introduction

Using features and classifiers to classify the image data into class labels has been popular in the recent years [8], [12]. Such techniques have been used with data obtained from a moving vehicle or robot for classifying outdoor data or to ascertain terrain navigability [3], [11]. However in the context of outdoor terrain classification the baseline classifiers can be improved to handle the spatial and temporal context of an image point in a stream of data. This is achieved through the partition based classification scheme that improved the baseline performance of the popular multiclass classifiers such as random forests and SVM. Such a partition based classification scheme was still not very well suited for handling changing concepts or concept drift that is associated with changing outdoor data sequences. A classifier that learns and adapts to changing concepts while retaining what was learnt about recurring concepts is thus entailed for the problem at hand.

Hence this paper also proposes an adaptive classification scheme based on a combination of a static and online partition based classifier. The output posterior probability of both the classifiers are fused to assign the eventual posterior probability of a class label. To facilitate online learning and adaptation, the online classifier is trained with a sparse but stable set of image patches that are stably tracked by an optical flow based tracker over a minimum number of time samples. The combined classifier thus adapts to and learns while preserving what has been learnt about recurring and stable concepts. This is seen in an improved error-rate of more than 20% for the combined classifier over and above the single monolithic classifier trained at bootstrap.

Such adaptive terrain classification systems find immense use in advanced driver assistance systems [14] and autonomous outdoor navigation. For instance, a mobile robot navigating outdoors, comes across various terrains such as soft, slippery, hard, smooth, rocky or undulating ones. The navigation strategy for the robot differs mainly based on the kind of terrain it traverses and the limits on its velocities vary according to these surfaces. An algorithm capable of prior judgment of the terrain provides the much needed time for the robot to adapt its velocity planner and thus becomes a vital cog in outdoor navigation systems.

II. Previous Work

Various methods have been proposed in literature for the problem of terrain classification. In particular Vibration-based methods [6], [13] (which use accelerometers, IMU) have been very successful. Yet, the main drawback of those methods is that they classify terrain only while the sensor attached to robot is traversing the terrain and not beforehand.

Camera-based methods follow a canonical form of using images from camera as training data along with lasers or stereo-rig for obtaining ground truth. Among the literature we surveyed, the work reported in [1], is closest to our problem. However it partially relies on time consuming texture features. Dima and Vandapal [3] trains separate classifiers on data from laser, infra-red camera and monocular camera and uses AdaBoost to combine the output. Bradley *et al.* [2] uses multi-spectral camera to detect chlorophyll content for recognizing grass and trees. Recent work includes, Blas *et al.* [9] employing pre-segmentation algorithm based on clustering using Local binary patterns. Vernaza *et al.* [10] uses Markov random fields framework. Procopio *et al.* [7] adds memory to the machine learning model by using ensemble of classifiers. They report an accuracy of around 90% on their datasets, but almost all of them detect only navigable region and do not characterize the terrain.

The current work is different from the above mentioned in that it explores the extent of scene interpretation ability vested in a single camera avoiding the use of lasers and other sensors in either generating initial class labels or as an aid to scene classification. This investigation is especially crucial since a camera is often much less expensive, compact, and not power hungry like laser range finders. Unlike many previous approaches, which deal with the problem of detection

Fig. 1: Sample frames from our dataset

of navigable regions, where the terrain characterization is neglected, our goal is to detect and characterize the terrain ahead into commonly observed terrains. The role of a partition based classification scheme enhancing the performance of base line classifiers has also not been reported before in literature and its further embellishment to learn and adapt to changing data are the contributions of this effort.

The remainder of this paper is organized as follows. First, in Sec. III we describe our experimental settings and baseline results. In Sec. IV we describe our novel partition-based-method, its experiments and results. In Sec. V we describe our Adaptive algorithm, its experiments and results. Finally conclusions and future work are provided in Sec. VI

III. Experimental framework

Terrain classification was modeled as a classification problem of pixels and smaller windows in the past [3], [11], where the important parameters were features, classifiers and datasets. In this section, we describe our features, datasets along with baseline results.

Features. For any learning based method, selecting meaningful features for the classification task is very important. We use popular RGB histogram [7], [11] and LBP histogram [9] as our features considering the computational cost and performance. We use the optimal weighted combination of these features that best suits the classifier.

Data sets. We experiment with three datasets in this study: our own dataset and two other datasets by Procopio *et al.* [7]. For collecting our data, monocular camera was mounted on the top of the vehicle, and videos were recorded at 7.5 fps.We set the camera to high aperture and high shutter speed, in order to minimize the artifacts caused by the moving camera like motion blur. We collected the data in and around a radius of 10km navigating at various speeds ranging from 0.2m/s to 4m/s. We observe that the data is challenging, as it contained wide variations in illumination. We also observed that the data varied from unpaved or damaged rural roads to paved urban roads. We collected 25 videos, each of 1 min. Figure 1 shows some of the sample frames from the recorded videos. Five different terrains were identified in the data collection. Data contains regions of road, muddy-road, rough-terrain, grass (Note that the class grass contains only traversible grass or very small plants, big plants and trees are considered obstacles.) and obstacles (which contains static objects like trees, rocks etc., and dynamic objects like moving vehicles).

Empirical evaluation For the empirical studies, we consider 200 images from our data set. We use 50% of the data for training and the rest for testing. We extract multiple, non-overlapping, patches of size 16×16 from these images. Thus we have around $2 * 185000$ patches (The number of patches in all the five classes are equal) for training and testing.

Baseline results Performance of selected features are evaluated on a set of popular and promising classifiers. The baseline classifiers which we consider in our experiments are Naïve Bayes(NB), K-Nearest Neighbor(K-NN), Artificial Neural Networks(ANN), Support vector machines(SVMs) and Random Forests(RF) [5]. Random forest is a classification algorithm that uses an ensemble of unpruned decision trees, each of which is built on a bootstrap sample of the training data using a randomly selected subset of feature space dimensions. Experiments were conducted by changing important parameters like number of epochs and number of nodes in the hidden layers in ANNs, number of trees and size of node in RF. In case of SVMs, we conduct experiments with linear SVM using 1 vs 1 multiclass classifier (SVM-L) and non-linear SVM (SVM-K). We observe that RF's outperformed all other classifiers because of its capability to handle large number of input variables and data samples [5]. Additionally RF classifiers are computationally efficient for training and testing, compared to SVMs. Therefore we choose RF as our classifier.

IV. Partition based algorithm

In the last section, we have mentioned that RF classifier is performing best among several baseline classifiers. In this section, we describe two enhancements for terrain classification. Initially we describe our partition based algorithm and several experiments which indicate that, the algorithm is robust and spatially smooth. Secondly we describe our label transfer method along with experiments showing that, it saves considerable amount of computation time.

The proposed algorithm partitions the training images(See Figure 2) and trains different classifiers on different parts of the image independently. This is repeated for partitions of different sizes. Training different classifier from different part of the image handles the problem of perspectivity of the imaging process, i.e., it learns the fact that near and far image patches show different textural characteristics. Also learning from fixed partition over several training images has two main advantages. The first advantage is that it helps the classifier to learn new facts about associativity of classes, such as occurrences of grass along with mud is more probable than that of grass along with tar road. The second advantage is that it helps the algorithm to be dependent upon the position of the partition of the image and thus learns the spatial context. By training a classifier from larger sized partitions, global properties of the class are learnt and as the size of the partition decreases, more local properties are learnt. Our algorithm is a generic framework that can be operated on any classifier.

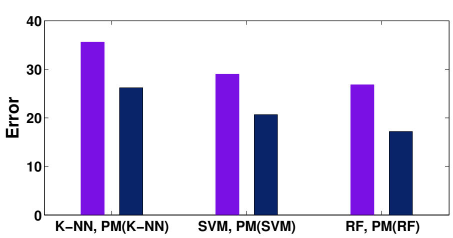
In training phase, as summarized in Algorithm 1 we build N classifier-sets using all the training images, let us call them $S = \{C_1, C_2, C_3, ...C_\mathrm{N}\}$. Note that a classifier-set C_i contains i^2 classifiers. To characterize the terrain of the given image, for each patch of the image, we get N labels from each of the N classifier-sets in S. From these N labels, most occurring label is declared as the final label of the patch.

A. Experiment 1: Comparison with baseline classifiers

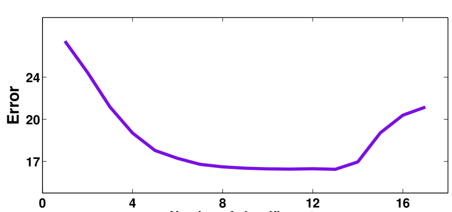
Figure 2a shows the percentage errors of our partition-based algorithm operating on baseline classifiers SVM and Random Forests. We observe that our algorithm always decreases the percentage errors by approximately 10%. This is an appreciable decrease in the percentage error. It also shows that our algorithm is generic, i.e., the algorithm improves the performance of classifier irrespective of the classifier chosen. To show the superiority of our algorithm across other databases, we conduct an experiment in which our partition-based algorithm operating over RF is tested on (i) Our dataset (ii) DS3A and (iii) DS3B datasets of Procopio *et al.* [7]. We report the percentage errors in first and second column of Table I, from the table, we observe that our algorithm compared to baseline RF classifier, decreases the percentage error by approximately 10% on all three datasets. We also observe that even without training on any of the images of DS3A or DS3B datasets, we get percentage error as low as 6.8%, the superiority of our algorithm is thus clearly evident.

B. Experiment 2: Effect on N or Learning cost

Figure 2b shows the effect of increasing the number of classifier-sets(N), note that as N increases the learning cost increases because one needs to train more classifier sets. Also, the classification performance decreases as N increases. We observe that as N increases, the percentages error initially decreases and then slowly increases. From the figure, we also observe that the optimal choice for N ranges in between 4 to 8, which has high efficacy and also low learning cost.



(a) Comparison of baseline classifiers with PM operated on them



(b) Error rates by using multiple classifier sets

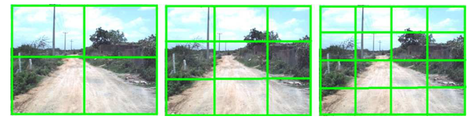


Fig. 2: Partitioning the image into 4,9 and 16 partitions.

Algorithm 1 Partition based algorithm(PM)

– *Training*

Goal: To build N classifier-sets
Input: M Training images, $S \leftarrow \emptyset$
for $k = 1$ to N **do**
  Partition training images into k^2 parts, $C \leftarrow \emptyset$
  for $p = 1$ to k^2 **do**
    Train a Classifier on p^{th} partition over all training images, call it KF
    $C \leftarrow C \cup \{KF\}$
  end for{ Now $C = \{KF_1, KF_2, ...KF_{k^2}\}$ }
  $S \leftarrow S \cup \{C\}$
end for{ Now S contains $\{C_1, C_2, ...C_N\}$ }

– *Characterize Terrain of given image*

Input: Image I
for all patches of Image I **do**
  $L \leftarrow \emptyset$
  for $i = 1$ to N **do**
    $l \leftarrow$ get the label of the patch from classifier set C_i
    $L \leftarrow L \cup \{l\}$
  end for
  Majority voted label from L is declared as final label of the patch.
end for

C. Experiment 3: Spatial smoothness test

In Table I, third and fourth columns show the smoothness-errors of RF and PM operated on RF(PM_RF), on three datasets. Smoothness-error is the difference between percentage errors before and after applying smoothing algorithm (MRF [15]) on the predicted labelled image. We observe that our algorithm has a negligible smoothness-error compared to RFs, which clearly shows that PM_RF itself is capable of characterizing the image smoothly in spatial context. Figure 3 shows the superiority of partition based algorithm over baseline RF classifier. We observe that the images labelled using our method are smooth in spatial context.

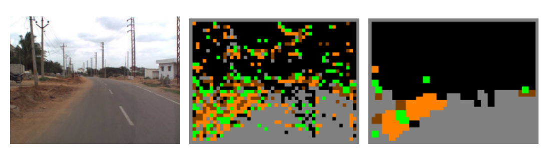


Fig. 3: Figure shows the test image, characterization by RF classifier and characterization by PM respectively

D. From Image to Video

Temporal label transfer. Most of the methods in literature deal with single image. They do not use the fact that they are dealing with a sequence of continuous video stream. When robot navigates through terrain, the camera captures sequence of frames. Any two consecutive frames have lot of common image regions. In order to characterize the terrain of the image using traditional machine learning based algorithm some kind of feature is extracted from each patch. The feature vector is fed to a classifier, which returns the label of the patch. Note that in this process, feature extraction is computationally expensive. In our case, when a new frame is captured by the camera, fast coarse optical flow [4] between the previously captured frame and current frame is calculated. For each patch of the new frame, if there is flow present, we transfer the corresponding label from the previous frame to the current frame, else feature is extracted from the patch and fed to our partition-based algorithm. In this way without even extracting features from the current frame, we can label considerable portion of the frame. However, while transferring the labels, few incorrectly classified labels are also transferred. Label transfer method is designed to transfer the labels without performing classification as transferring labels is relatively low in computational cost.

We conduct an experiment, to check (i) how much amount of incorrect labels are transferred and (ii)what portion of the image can be labelled by just using temporal label transfer. The average percentage of image that is labelled correctly over testing images is reported in fifth and sixth column of Table I. We observed that by just using temporal label transfer, we can label approximately 40% of the image on three datasets with significantly lower percentage error. This saves around 40% of the total time taken (which includes feature extraction time and classification time), such a reduction in time is crucial in real time systems like robots.

V. ADAPTIVE METHOD

The canonical offline or memory-less classifiers tend to perform poorly in outdoor environments because these environments contain huge variations in illumination. One of the solutions to this problem is to train the algorithm on all possible variations of illuminations, which is impractical.Also, In general, increasing the amount of training data drastically, decreases the performance of classifier. These motivate us for developing a terrain classification scheme, that is capable of classifying the terrain in dynamic environments. Previous laser based solutions [7] for this problem are appreciable, but our aim is to classify terrain using only monocular camera, where collecting online ground-truth is impossible. In this section, we describe our scheme for this problem that would enable the robot to adapt to unseen images.

In the proposed algorithm (summarized in Algorithm 2 in page 4), let us denote the current frame with I_i. The previous P frames would be I_{i-P},I_{i-P-1}...I_{i-1}, which are already labelled by our scheme. Using the previously computed flow between successive frames in Section IV-D, we track the patches from previous frames at an interval

Dataset	RF	PM	RF	PM	AVG	Err
O	26.8	17.2	08.7	01.0	35.5	05.6
P-A	18.2	07.9	06.9	00.6	42.3	04.3
P-B	18.9	06.8	05.2	00.4	45.1	04.3

TABLE I: 1^{st} and 2^{nd} column represents percentage errors of RandomForest(RF) and our partition based algorithm(PM). 3^{rd} and 4^{th} column represents smoothness-error, which corresponds to experiment-3. 5^{th} and 6^{th} column represents the percentage of images, that were labelled just by using Temporal-label-transfer method in Section IV-D, where AVG: Average of percentages of portion of labels that are transferred over sequence of 100 images and Err: Error in label transfer

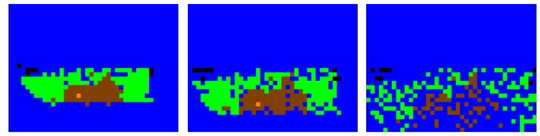


Fig. 4: Tracked patch-labels across three frames.

Algorithm 2 Adaptive algorithm

– Training

$I_c \leftarrow$ current image that needs to be classified.
$P \leftarrow$ Number of previous frames to use.
$stepSize \leftarrow$ Number of frames from which the patches are to be tracked.
$previousFrames = \{I_{c-P}, I_{c-P+1}, I_{c-P+2}, ...I_c\}$
$newTrainingData \leftarrow \emptyset$
for $i = 1$ to $\lfloor P/stepSize \rfloor$ **do**
  $j = c - i * stepSize$
  Track patches from the previous frames $\{I_{j+1}, I_{j+2}, ...I_{j+stepSize}\}$.
  for all tracked patches p_j **do**
    $Label(p_j) \leftarrow$ {Most repeating label among $stepSize$ labels}
  end for
  Update $newTrainingData$ with tracked patches and their corresponding labels.
end for
$onlinePMmodel \leftarrow$ Get the model from Partition-based-method trained on $newTrainingData$.

– Characterize Terrain of given image

Input: Image I_c
for all patches of Image I_c **do**
  $P1 \leftarrow$ the posterior probabilities from offlinePM-model
  $P2 \leftarrow$ the posterior probabilities from onlinePM-model
  $P1 = P1 + P2$ {Fuse the results}
  Label corresponding to maximum probability in P is declared as final label of the patch
end for

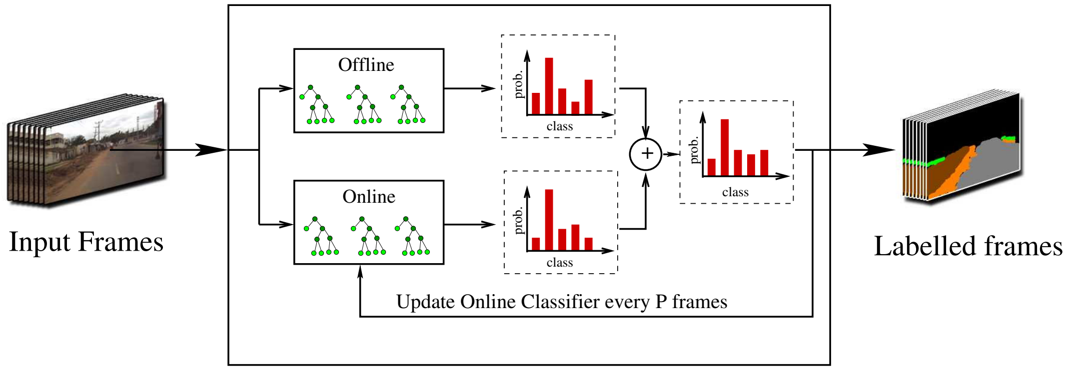


Fig. 5: Block diagram of the proposed scheme

of K frames. We use $K = 5$ in our experiments. For example, Figure 4 shows the tracked patches from three successive frames. These tracked patches across frames slowly vary in their illumination and perspectivity. For each of the tracked patches, we have K labels associated from the K frames. We label the tracked patches accurately by selecting the most repeating label from the K labels. Note that, here the training relies on off-line classifier, even if the off-line classifier classifies wrongly, the final labels are superior because we choose the most repeating label from K labels. We train another Partition-based classifier on these tracked patches on previous P frames, we call this classifier as online-partition-based-classifier. We update the online-partition-based-classifier every P frames.

To characterize the terrain of the current frame, the posterior probabilities of Offline partition based classifier and online-partition-based-classifier are added (see Figure 5 in page 5). In case, two of the posterior probabilities are close, we choose the label that is most repeated with in the neighborhood of the patch.

Implementation details. In our experiments we train online-partition-based-classifier every 200 frames, note that while training the captured frames are classified independently. Hence online training and classification can be executed in parallel. Also using RFs internally adds another advantage. In RF, the final posterior probability is fused result of several posterior probabilities of several trees, here each tree can be used independently and hence can be executed in parallel. These advantages make our algorithm parallel and can be implemented efficiently using GPUs.

A. Performance Gain Due to Adaptive Classifier

In this section we show both by quantitative and qualitative experimental results the advantages of having an online classifier. Quantitatively we show decrease in errors on 6 data sets, including two publicly available data sets. Qualitatively we show those portions in the image where the adaptive classifier has corrected wrongly classified patches by the Partition method. We also show results from an experimental run where the vehicle reaches the location from where it started its journey.

1) Quantitative and qualitative results: Table II shows the percentage errors of Offline-partition-based-method and

Dataset	**PM**	**Adaptive**	**Error-rate**
O-A	18.2	12.7	30.2
O-B	20.2	15.9	21.2
O-C	17.0	13.3	21.7
O-D	17.5	16.1	07.9
P-A	07.9	05.3	32.9
P-B	06.8	06.1	10.2

TABLE II: Comparison of Adaptive algorithm with Offline-partition-based-method

Adaptive algorithm on 6 sequences in columns 2 and 3. Since the Offline-partition-based-method already achieves a reasonably low percentage errors, further improvements over Offline-partition-based-method by Adaptive algorithm can be portrayed in terms of rate of decrease in error, which is $Error\ rate = \frac{\%\ error\ of\ PM\ -\ \%\ error\ of\ Adaptive\ algorithm}{\%\ error\ of\ PM}$. The error-rates were presented in column 4 of Table II. The first four rows of the table correspond to 4 sequences of our dataset. In these sequences, the robot is navigated continuously until 800 frames were captured. Adaptive algorithm is applied on these 4 sequences independently, where the online-classifier is updated every 100 frames. The last two rows show the percentage errors on datasets by Procopio [7], since their data-set is a sequence of only 100 frames, the online-classifier is updated every 20 frames. 20 randomly picked images from each sequence were used for testing. We observe that the Adaptive algorithm has a huge decrease in error-rate of more than 20% on almost all the sequences. This clearly shows the superiority of the proposed scheme. The adaptive algorithm was tested on our dataset, we observed that dataset contains gradual and sometimes fast gradual appearance changes but not sudden appearance changes. Figure 6 shows some of the test images marked with the red-colored-patches from our dataset. They represent the labels that are correctly labelled by Adaptive algorithm, which are wrongly labelled by offline Partition-method.

2) Closed loop test: The closed loop test is a means to evaluate if the performance of the adaptive algorithm improves over time, the knowledge embedded in the classifier is not static and has adapted with passage of time. The improved performance comes by exploiting the data that comes on the fly, while simultaneously not forgetting what was learned at bootstrap. At the beginning of the run the robot has learned based on the offline dataset representing bootstrapped

Fig. 6: Test images marked with red-colored-patches, representing the labels that are correctly labelled by Adaptive algorithm but wrongly labelled by offline Partition method.

knowledge. As the run progresses the knowledge is expected to be enhanced. By showing improved performance upon reaching the starting location after a run of more than 2km we verify that the objective of learning without forgetting the past is realized.

In this experiment, we test our Adaptive algorithm in a closed loop path (see Figure 7 in page 6)i.e., the Adaptive algorithm is applied on data which was collected by navigating the robot on the same road twice. 20 random images from each loop at approximately same locations were used for testing. Note that not even one of these images were used in the initial offline training dataset. We observe that the mean error on the round-1 is 16%, where as the mean error for round-2 was observed to be 13%. The decrease in percentage error was observed mainly because, the adaptive algorithm slowly adapts itself to the new environments. Second row of the Figure 7 shows the test image along with the predicted labelled images from the first and second loops. We observe that the wrongly labelled mud(orange) patches in first loop are being correctly labelled in the second loop.

VI. Conclusions and future work

This paper presented a novel partition-based algorithm for classification of outdoor terrains using monocular camera. The proposed algorithm is generic and enhanced the percentage error of base-line classifiers by approximately 10%. The partition-based algorithm was extensively tested on our dataset and on other publicly available datasets and its efficacy established. Partition-based method was extended to Adaptive algorithm by learning from the data by fruitfully exploiting the data that was obtained on the fly. Concepts may drift over time, offline classifiers may not adapt to these drift as effectively as a classifier that also adapts online. The adaptive algorithm was tested on several data sets, where an average decrease in error rate of around 20% was observed to portray its advantages. Further we show results where a vehicle upon coming back to the same starting point after traversing a loop of more than 2km improves its performance during the second traversal of the loop. This demonstrates

Fig. 7: 1_{st} row: Path navigated by robot in a closed loop, marked in green color. 2_{nd} row: Test image with predicted labelled images from the first and second loops.

that the adaptive classifier is able to adapt to changes that occur during a traversal while holding on to what was learned at bootstrap or before the commencement of navigation. The future scope of our work includes much better processing of the video data using complex temporal clues along with fusing geometric and appearance clues.

References

[1] D. Helmick A. Angelova, L. Matthies and P. Perona. Fast terrain classification using variable-length representation for autonomous navigation. In *CVPR*, 2007.

[2] Ranjith Unnikrishnan Bradley and J. Bagnell. Vegetation detection for driving in complex environments. In *ICRA*, 2007.

[3] Nicolas Vandapel Cristian Dima and Martial Hebert. Classifier fusion for outdoor obstacle detection. In *ICRA*, 2004.

[4] R Chellappa H Liu and A Rosenfeld. Fast two-frame multiscale dense optical flow estimation using discrete wavelet filters. In *Journal of the Optical Society of America*, 2003.

[5] Breiman L. Random forests. In *Machine Learning*, 2001.

[6] P. Leang and B. Bhanu. Learning integrated perception-based speed control. In *ICPR*, 2004.

[7] Jane Mulligan Michael J. Procopio and Greg Grudic. Learning in dynamic environments with ensemble selection for autonomous outdoor robot navigation. In *IROS*, 2008.

[8] Florent Monay, Pedro Quelhas, Jean-Marc Odobez, and Daniel Gatica-Perez. Contextual classification of image patches with latent aspect models. *Journal on Image Video Process.*, 2009:1–20, 2009.

[9] A Sundaresan Morten Rufus Blas, Motilal Agrawal and Kurt Konolige. Fast color/texture segmentation for outdoor robots. In *IROS*, 2008.

[10] Ben Taskar P. Vernaza and Daniel D. Lee. Online self-supervised terrain classification via discriminatively trained submodular markov random fields. In *ICRA*, 2008.

[11] A. Talukder R. Manduchi, A. Castano and L. Matthies. Obstacle detection and terrain tlassification for autonomous off-road navigation. In *IROS*, 2005.

[12] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, 2005.

[13] N. Fechner M. Stark Weiss C. and A.Zell. Comparison of different approachs to vibration-based terrain classification. In *European Conf. on Mobile Robotics*, 2007.

[14] Naito T. Yamaguchi K., Watanabe A. and Ninomiya Y. Road region estimation using a sequence of monocular images. In *ICPR*, 2008.

[15] Ting Chuen Pong Zoltan Kato and John Chung Mong Lee. Color image segmentation and parameter estimation in a markovian framework. In *Pattern Recognition Letters*, 2001.