Detection of Roadside Vegetation Using Features from the Visible Spectrum

I. Harbaš and M. Subašić
Faculty of Electrical Engineering and Computing, Zagreb, Croatia
iva.harbas@fer.hr

Abstract – Detection of vegetation in images is a common procedure in remote sensing and is commonly applied to satellite and aerial images. Recently it has been applied to images recorded from within ground vehicles for autonomous navigation in outdoor environments. In this paper we present a method for roadside vegetation detection intended for traffic safety and infrastructure maintenance. While many published methods for vegetation detection are using Near Infrared images which are particularly suitable for vegetation detection, our method uses image features from the visible spectrum allowing the use of common on-board color cameras. Our feature set consists of color features and texture features. One of our specific goals was to identify a useful texture feature set for the problem of vegetation detection. Based on the feature set, the detection is implemented using a Support Vector Machine algorithm. For training and testing purposes we recorded our own image database consisting of different images containing roadside vegetation in various conditions. We are presenting promising experimental results and a discussion of specific problems experienced or expected in real-world application of the method.

Keywords **- image analysis, image processing, vegetation detection, traffic safety.**

I. INTRODUCTION

Detection and analysis of the state of vegetation from images is considered very important in the field of remote sensing where it is used for detecting green areas on Earth and detecting changes caused by urbanization. Detection of stressed vegetation [2], [10], [5] and other similar applications are aimed at raising environmental awareness and improving our ecological footprint.

Vegetation detection in other areas besides remote sensing, e.g., in robotics, is very young and unexplored. The need for detecting vegetation in this area arose when autonomous vehicles were used for forest exploration. Vegetation is not a static obstacle and all of vegetation-like obstacles don't need to be avoided (i.e. a vehicle can go over tall grass) [9]. Hence, it was necessary to detect vegetation and decide if the vehicle can drive over that obstacle or should it be avoided. Increased use of autonomous vehicles for off-road navigation is the main reason for new research efforts in detection and classification of vegetation [1], [3], [6], [7], [8].

In this paper we present a pixel-based method for roadside vegetation detection from within a moving vehicle intended for traffic safety and infrastructure maintenance (e.g., mowing the grass along the road).

Many published methods for vegetation detection are using Near Infrared (NIR) images where vegetation has distinctive properties. Our method uses image features from the visible spectrum, imitating how people detect vegetation using color and texture features and allowing us the use of a common on-board color camera. The benefit of this approach is that the same camera can be used for other computer vision tasks which require color images recorded from within a moving vehicle. It should be made clear that many publications mentioning successful vegetation detection often indicate some very specific species of vegetation or a very limited dataset. And these approaches are then just used for robots operating in a specific environment or for satellite images but not for roadside vegetation detection using a camera mounted on a vehicle. Due to different goals, those methods cannot be directly compared to our method for detection of vegetation.

A brief overview of some proposed methods for vegetation detection is given in the second section of this paper. Our approach is introduced in the third section where feature selection and classifier training is described. Our methods results will be presented in the fourth section. Conclusions and future work are presented in the last section of this paper.

II. RELATED WORK

People easily detect vegetation on the basis of typical visual characteristics such as color, texture and shape. Often researchers try to mimic the human visual system when developing methods for automatic detection of vegetation. Studies have shown that there is valuable information for vegetation detection in the spectrum invisible to the human eye and therefore, systems for detecting vegetation can be divided into two groups:

- Systems based on the visible spectrum
- Systems based on the invisible spectrum.

Systems based on visible spectrum use color and/or texture features. Using only color as a feature can give good results but objects of the same color as vegetation (e.g., green car) are falsely detected. Such objects can be distinguished from vegetation using additional texture features. A green car does not have a strong texture while the texture of vegetation is unstructured and turbulent.

This research has been partially supported by the European Union from the European Regional Development Fund by the project IPA2007/HR/16IPO/001-040514 VISTA - Computer Vision Innovations for Safe Traffic.

Some researchers use LiDAR (*Light Detection And Ranging*) sensors which give information about the 3D structure of the scene for calculating texture features. Nguyen et al. use a sliding cube across the 3D-point cloud in space to capture the local point statistics features. For every sliding cube they calculate a positive definite covariance matrix and extract the principle components (eigenvalues and eigenvectors) of that matrix. The relative differences of the three eigenvalues indicate a spatial structure. They segment the 3D-point cloud into three classes: surfaces, linear structures and porous volumes (foliage grass, tree canopy) [6], [9]. It can be argued that using only 3D-data can't result in a robust detection of vegetation because no color information is used. Therefore, Nguyen et al. address the 2D-3D fusion approach which combines 2D-3D information for vegetation detection [6]. For 2D features they use mean and standard deviation of brightness and color, and the difference of histograms [6]. These features are used to train a SVM (*Support Vector Machine*). The problem with this approach is mapping the 2D and 3D information because they are obtained by two different sensors. For this in [6] they use a coarse calibration method. This approach is time consuming and it should be used when time is not a criteria [9]. Liu et al., in a similar way, are using a combination of 2D and 3D features. The feature vector consists of the height calculated from the 3D space obtained with LiDAR, and the H and S components obtained by converting RGB (*Red, Green, Blue*) to HSV (*Hue, Saturation, Value*) color space [3].

Except for navigation, detection of vegetation is also used to improve the quality of video, or TV images. In [4] and [11] authors used texture features and color components from the YUV color space as color features. After detection of grass in video sequences they enhance the image by changing the color or brightness of pixels in that segment.

Using features from the visible spectrum has some problems (leaves change color, green cars, different illumination scenes, etc.) and that is why researchers started using features from the invisible spectrum. This idea was present for many years in remote sensing where researchers found that chlorophyll rich vegetation has a high reflectance in NIR part of the spectrum and can clearly be distinguished from the sky and other objects. Using this property of vegetation many different features can be calculated which are called Vegetation Indices. Mostly used is NDVI (*Normalized Difference Vegetation Index*) [8].

Although NDVI has been successfully used in remote sensing the work of Bradley et al. [1] has shown that there is a drastic difference between the view-points of a satellite and an autonomous ground vehicle where there are more problems typical for on-ground recordings such as shadow, shining and underexposure effects. Taking that into consideration and wanting to avoid time consumption needed for calibration and LiDAR scanning Nguyen et al. developed a modified vegetation index MNDVI [9] which they use as a feature for vegetation detection and all they need is NIR intensity and color information. In [8] a combination of vegetation indices with color and texture features is used for vegetation detection. First, by setting a threshold for NDVI and MNDVI they obtain pixels rich in chlorophyll. These pixels are seeds for spreading algorithm that follows. For each seed pixel distance in color and texture between the seed pixel and his neighbors is calculated.

Systems based on invisible spectrum require equipment for recording NIR images and systems based on visible spectrum that use LiDAR for feature extraction need an additional 3D scanner.

We require no additional sensors but a single color camera for data acquisition so the features that will be used for detection are from the visible spectrum only.

III. METHODOLGY

In this section we describe our method for detecting roadside vegetation using only features from the visible spectrum which does not require any additional equipment except a common color camera mounted onboard. Our method for vegetation detection comprises of three steps:

1. extraction of selected features,
2. pixel-based classification and
3. postprocessing.

To the best of our knowledge there is no publicly available database that could be used for our application. That is the reason we made our own database for testing and all results presented in this paper were obtained using this database. The database was recorded using a High Definition Camcorder Canon XF100 from a moving vehicle. The videos have been recorded in daylight at different times of the day to include different lighting conditions. The database includes different traffic scenarios including roadside vegetation in various conditions. Currently our database contains 270 images extracted from the recorded video sequences, each having 1920x1080 resolution. These images were manually segmented into vegetation and non-vegetation regions.

The intended use of our method requires that exact contours of each vegetation region are detected so we perform per-pixel classification into vegetation or non-vegetation classes.

A. Feature extraction

Features that we extract from color images to detect vegetation are color and texture features.

1) Color features

Human perception of color is one of the most important visual elements which help us visually recognize different objects. Color is an important descriptor of vegetation because vegetation doesn't have a specific shape but is usually represented by green, orange or yellow color. Transferring this human ability to a computer algorithm is far from simple. One major problem is the change in intensity and color in different light conditions. This usually doesn't pose a problem for the human visual system but a change in luminance can cause a significant change in image features.

Some authors consider using different color spaces for color features (e.g., HSV [3], YUV [4], [11]), while others take into account the mean and standard variation values of intensity and color [6]. In [3] they use only H and S components, ignoring the luminance component V so their feature set would be less sensitive to light changes in the scene.

Taking this into account the first step was deciding which color features would be most suitable for our problem. Three color spaces commonly used in published vegetation detection methods were used in our test: RGB, HSV and CieLAB. We didn't test YUV color space because YUV is used for a specific analog encoding of color information in television systems and the only reason it is used in [4] and [11] is because they used it for TV image enhancement. CieLAB was added for testing because it is designed to mimic human perception of color. Additionally, we tested only the H and S components excluding the lightness from HSV and only A and B components excluding the *L* component from CieLAB.

We used five different feature vectors containing different color features to train a SVM classifier with linear kernel. This classifier was used only in this stage of testing to evaluate how appropriate each feature vector is for our problem.

To properly evaluate how each of our color features will generalize to an independent data set we use 10-fold cross-validation to measure the performance and to validate our results. The results are shown in Table I.

From these results we conclude that excluding the lightness component from the feature set does not affect the results drastically. Slightly better performance can be seen when using the lightness component with color information, i.e., better accuracy is achieved using HSV and LAB then HS and AB features respectively. The second conclusion we draw from Table I. is that RGB and LAB color spaces perform similarly and are giving better results compared to HSV. Vegetation color should theoretically be green in HSV color space under most different environment conditions. In reality, this is not always true for scenes containing sky or varying lighting conditions (shades, overexposure, and underexposure). Low intensity of the value V in HSV color space tends to turn the color of image to red, red brown, etc [6]. And that is why using HSV color space has more false positive errors where red objects in scenes are detected as vegetation. On the basis of conclusions previously presented, we decided to use RGB color components as color features.

2) Texture features

TABLE I. COMPARISON OF DIFFERENT COLOR SPACES USED FOR VEGETATION DETECTION

	Feature vector				
	RGB	***LAB***	***HSV***	***HS***	***AB***
Accuracy	92,6873	92,767	87,3196	87,1123	91,1518

Simple color features lack discriminative power required for our problem where non-vegetation objects may have colors similar to vegetation like illustrated in **Fig. 2 b.**. Additionally when dry vegetation, which takes on shades of yellow and red, is added to the training set, the number of falsely detected objects increases. To solve this problem we added texture features to the feature vector.

There are many different methods for calculating texture features in images and choosing one is a difficult task. Knowing that vegetation is diverse and that vegetation regions in images contain more information than homogeneous surfaces, we decided to use entropy as a texture feature. Entropy is a statistical measure of randomness and it can be described as a measure of the amount of disorder in a system. For images it can be expressed as a spread of states (gray levels) which the individual pixel can adopt. If pixels in an image, or in a block of an image, have the same values the entropy is zero and if an image (or a block) contains pixels changing in unexpected ways the entropy is high. We expect that vegetation region in an image have high entropy.

The entropy *H* of an image is defined as [12]:

$$H = -\sum_{k=0}^{M-1} p_k \log_2 p_k \tag{1}$$

Where *M* is the number of gray levels and p_k is the probability associated with gray level *k*.

B. Feature ranking

Using the entropy of the grayscale image did not yield better performance so calculating entropy in different color spaces was considered. Color spaces taken into consideration were the same ones used for testing color features. There are a lot of possible combinations of color features and entropy in different color spaces that we could use as a feature vector for testing. To find the best one we used feature ranking as a filtering method where all these features are ranked, and based on the calculated rank we choose the best one.

To understand the influence of features on the system or even if the number of features is to large feature ranking is a good method to get baseline results and to assess features individually [13]. We created a feature vector that contains our selected three color features and 9 texture features that are entropy calculated for R, G, B, H, S, V, L, A and B components of the corresponding color space models. Total number of features is 12. Twelve features is not too much to handle for any kind of feature selection, but feature ranking is a good filter method and a good preprocessing step independent of the choice of the predictor [13]. The goal of using any kind of filtering of features is to eliminate possible outliers. Ranking was done using Wilcoxon rank sum test which is a nonparametric test for equality of population medians of two independent samples [14]. A nonparametric test was used because they don't assume normal distributed classes and our samples were tested and showed that they are not normally distributed.

Variable ranking makes use of a scoring function computed from the input variables (calculated features) and output variables (class assignment). For this test the scoring function is the rank sum statistic calculated for every feature in the feature vector. By convention, we assume that a high score is indicative of a valuable variable [13].

This test was repeated 10 times so that statistical significance can be established. For each of the ten different iterations, R, G and B features were always the three top ranking features followed by entropy calculated for saturation (S component of HSV color space). This is the only entropy that was mostly top ranked while others varied drastically from test to test. For this reason we added only this entropy to our feature vector.

C. Training the classifier

SVM is a well known supervised learning algorithm used for classification and in this paper it is used to discriminate vegetation and non-vegetation. An SVM is trained with a set of training examples each marked as belonging to one of two classes, in this case, vegetation and non-vegetation. Positive training examples were selected randomly from hand segmented vegetation regions in images and negative examples were also selected randomly but we made sure that different problematic objects (green, red, and yellow cars, green T-shirt of a pedestrian, yellow stripes on the road, etc.) are present in the negative set so the classifier would be trained on these examples too.

Selecting the training data is a very important step because all further classification is dependent on the trained model. Using only green vegetation for training makes a classifier that detects green vegetation but doesn't detect vegetation that is yellow or red (dry vegetation) (**Fig. 1.**) or different species of vegetation. Adding dry vegetation to the training set makes a classifier that detects dry vegetation but it detects more false objects as vegetation including yellow and red objects (and not only green objects).

After training examples are selected the SVM training algorithm builds a model that assigns new examples into one category or the other [15].

Our database contains 270 images of resolution 1920x1080 which presents 1920*1080*270 pixels available for training and testing the classifier. A smaller sample needs to be selected because training an SVM with a large number of samples can lead to overfitting and a very complex model can be built which can be time consuming during classification.

There are various recipes for calculating the required sample size which require knowledge of the variance or proportion in the population, the maximum desirable error, as well as the acceptable Type I error risk (confidence level).

It is possible to construct a table [16] that suggests the optimal sample size – given a population size, a specific margin of error and a desired confidence interval. Based on the table from [16] we found that the best sample size is 4000 (2000 samples representing vegetation and 2000 samples representing non-vegetation). These 4000 samples are chosen randomly. We have tested the algorithm using a greater number of samples but it only increased the computational time and it didn't improve accuracy.

a) b)

Fig. 1. Detection using: a) only green vegetation for training, b) all vegetation for training

D. Postprocessing

Classification is done for every pixel in an image. The resulting image with two classes marked often has some pixels marked as vegetation inside a non-vegetation region and non-vegetation pixels inside a vegetation region. It is safe to assume that pixels, or groups of pixels surrounded by vegetation also belong to vegetation. Also, solitary pixels or small groups of pixels surrounded by non-vegetation is often non-vegetation but is misclassified because it is too similar to underexposed vegetation (shadows under cars or parts of asphalt) or it has yellow reflectance (dry vegetation is mostly yellow). These gaps and solitary pixels can be removed after classification using a combination of morphological operations on the image.

In the postprocessing stage of our method we use morphological opening. The morphological open operation is an erosion followed by a dilation, using the same structuring element for both operations [17]. After opening further improvement was done by additionally removing solitary groups of pixels of certain size (for which we presume that are misclassified non-vegetation pixels) followed by filling up patches of certain size misclassified as non-vegetation (surrounded by vegetation).

IV. Experiments and Results

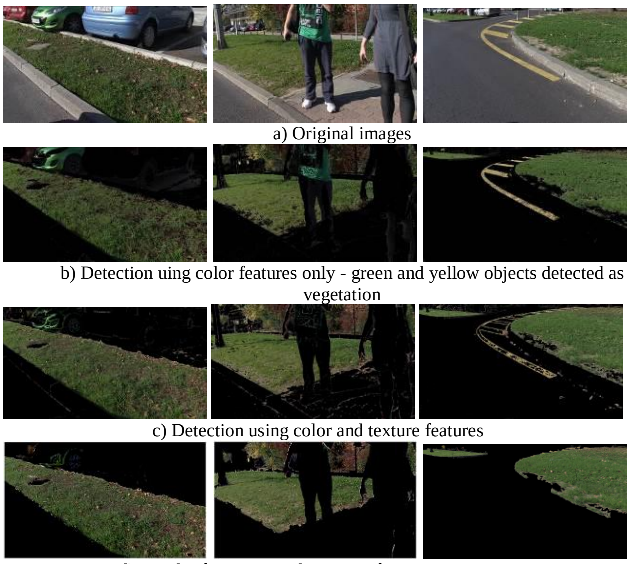
To find the optimal parameters for our method we tested several options for every aspect and choose the ones that gave the best accuracy.

Entropy is calculated for every pixel taking into account its neighborhood. The neighborhood size is a changeable variable. We tested four block sizes: 7x7, 9x9, 11x11, and 13x13. Experiments showed that the optimal neighborhood size is 9x9 (the smaller one showed worse accuracy, while the bigger ones did not improve accuracy). For training and classification, beside the linear kernel, we tested quadratic and radial kernels to take advantage of SVMs kernel trick and Cover's theorem. The best classification accuracy was obtained using radial basis SVM. For postprocessing, three variables have to be set: the structuring element, size of patches to remove, and size of patches to fill. We have used a circular structuring element with 5 pixel radius. At this step we presume that vegetation parts of an image are well connected and those parts in the image are large so we remove the ones that are less than 3000 pixels and filling was done for patches less than 500 pixels big. The small value of 500 was used because there are parts of roadside vegetation that have utility shafts that are small and any bigger size of 500 would misclassify these parts as vegetation (**Fig. 2 b.** and **c.**, **Fig. 3 b.** and **c.**).

Adding texture features improved detection in problematic images. With these features the green objects in **Fig. 2 a.** are much better classified as non-vegetation seen in **Fig. 2 c.** compared to **Fig. 2 b.** where only color features were used. Also, the yellow stripes on the road were falsely detected as vegetation when only color is used as a feature (**Fig. 2 b.**) but adding texture improved detection (**Fig. 2 c.**). In **Fig. 2 c.** misclassified solitary pixels and groups of pixels can be seen which we improve using postprocessing (**Fig. 2.d.**).

For final testing of our method we divided our database into training and testing set. Training set consists of 180 images which is 2/3 of the total number of images in the database, and from this set we are randomly choosing 4000 pixel samples for training. The remaining 90 images were the testing set, and all their pixels were testing samples. Because we select our training samples from the training set randomly we must repeat the selection process to exclude the possibility of choosing “the perfect” training samples and presenting statistically insignificant accuracy. We ran 10 iterations of training and testing, every time randomly selecting the training set and achieved average accuracy of pixel classification is 94.995%. We used high definition images in our training and testing and because of that the classification is somewhat time consuming but at this stage of our research that is not a concern for us as we are not focused on real-time performance.

Some results of vegetation detection using the method presented in this paper are shown in **Fig. 3**. In **Fig. 3 a.** and **b**. more examples of green objects in scenes correctly detected as non-vegetation can be seen. In **Fig. 3 c. – e.** we see good detection in different traffic scenes, while good performance in detecting vegetation in shade can be seen in **Fig. 3 f. - h.** Also good detection is presented in underexposure and overexposure conditions (**Fig. 3 i.** and **j.** respectively). In **Fig 3 k.** and **j.** we can see poorer detection results. In **Fig 3 k.** parts of the yellow car are still detected as vegetation even though we included these examples in the training set. This color is too similar to dry vegetation examples in the training set and because entropy is high over the edges the classifier decided that this is vegetation. In image **Fig 3 j.** vegetation that is far from the camera is not detected due to a lack of details in that part of the image, combined with a color deviating from green. Parts of the parked dark car are also detected as vegetation due to green reflections on its hull.



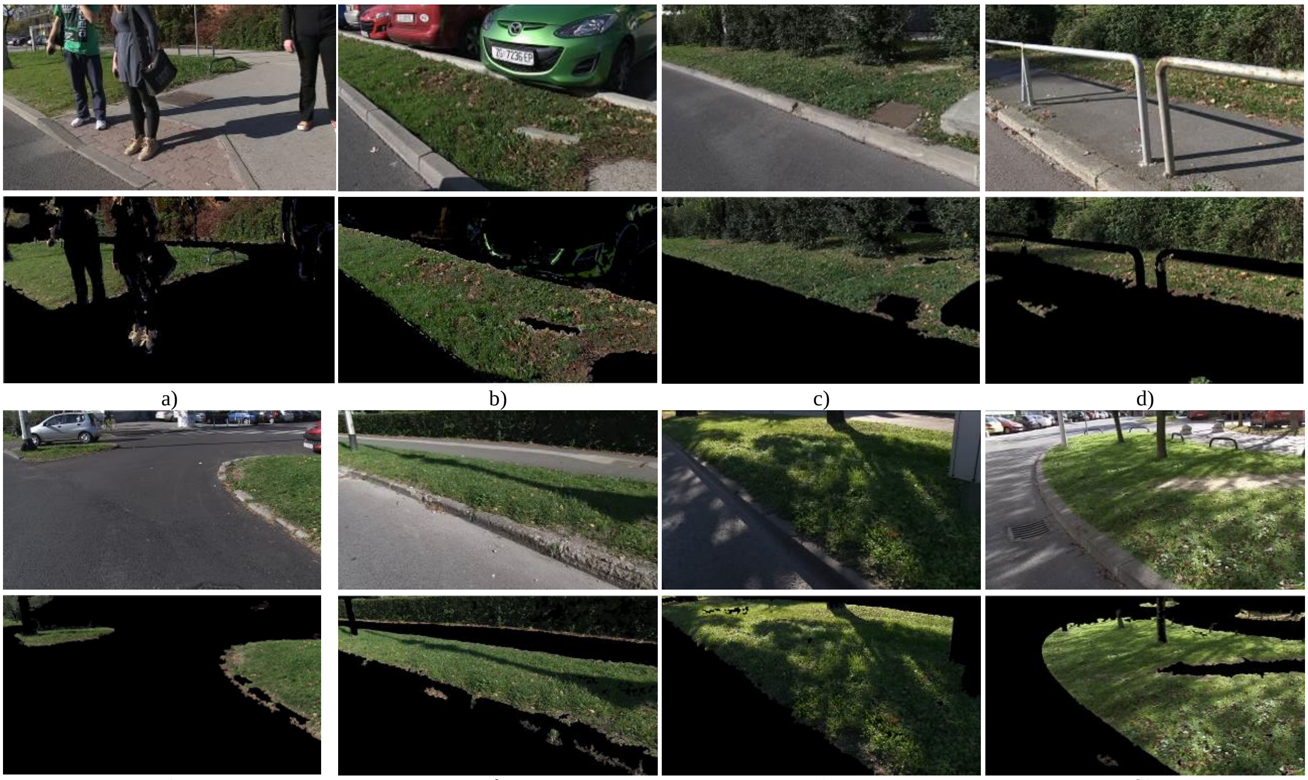
a) Original images

b) Detection uing color features only - green and yellow objects detected as vegetation

c) Detection using color and texture features

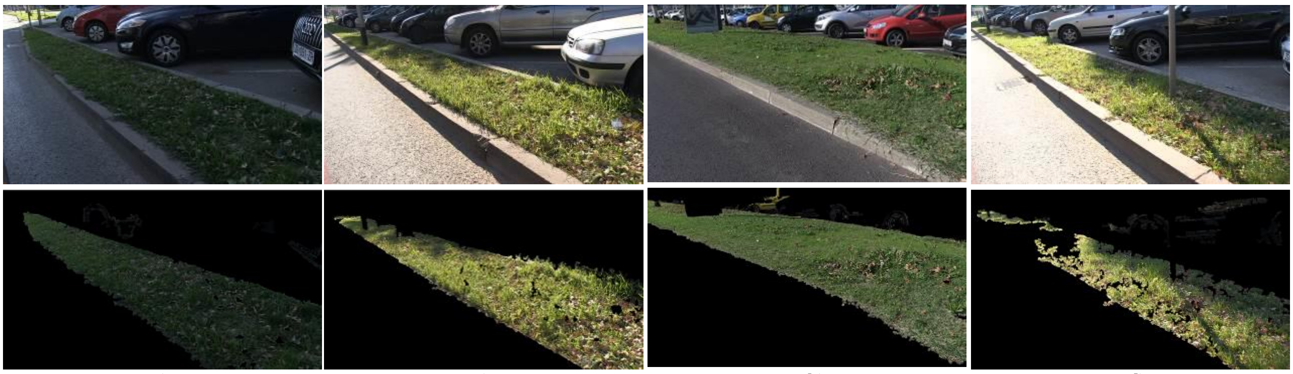
d) Result of vegetation detection after postprocessing

Fig 2. Results of vegetation detection for problem images



a) b) c) d)

e) f) g) h)



i) j) k) l)

Fig. 3. Results of detection using the method presented in this paper (top – original image, bottom – result of detection)

V. Conclusion

In this paper we present our method for detecting roadside vegetation using only features from the visible spectrum. This approach allows us to use common multi-purpose color cameras, and we are using both color and texture features. Based on our experiments, RGB color space proved to be the optimal choice for our color features. In the similar manner, we have selected entropy as our texture feature. We require a texture feature because color features alone were insufficient to solve problems with common situations where green non-vegetation objects occur in images. Entropy has mostly solved the problem of false positive green objects, but to further improve detection results we have implemented a post-processing step based on morphological operations. We have compiled our own data set for training and testing, with special attention paid to the training data set. It was compiled from random images and from several especially problematic images. The classifier we have used is SVM with radial kernel. We have obtained promising experimental results, and for our future work we plan to implement complex texture features from the frequency domain to increase accuracy. Real-time execution was not in the focus of presented research so current version of our vegetation detection method is not optimized for speed.

References

[1] D.M. Bradley, R. Unnikrishnan, and J. Bagnell. "Vegetation detection for driving in complex environments". In Robotics and Automation, 2007 IEEE International Conference on, pages 503–508, 2007.

[2] E. J M Carranza and M. Hale. "Remote detection of vegetation stress for mineral exploration". In Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International, volume 3, pages 1324–1326 vol.3, 2001.

[3] Liu da xue, Wu Tao, and Dai Bin. "Fusing ladar and color image for detection grass off-road scenario". In Vehicular Electronics and Safety, 2007. ICVES. IEEE International Conference on, pages 1–4, 2007.

[4] Stephen Herman, Johan Janssen, Erwin Bellers, and James Wendorf. "Automatic segmentation-based grass detection for real-time video", December 14 2004. US Patent 6,832,000.

[5] P. Kempeneers, S. De Backer, W. Debruyn, P. Coppin, and P. Scheunders. "Generic wavelet-based hyperspectral classification applied to vegetation stress detection". Geoscience and Remote Sensing, IEEE Transactions on, 43(3):610–614, 2005.

[6] D. V Nguyen, L. Kuhnert, T. Jiang, S. Thamke, and KD Kuhnert. "Vegetation detection for outdoor automobile guidance". In Industrial Technology (ICIT), 2011 IEEE International Conference on, pages 358–364, 2011.

[7] D. V Nguyen, L. Kuhnert, S. Thamke, J. Schlemper, and K. D Kuhnert. "A novel approach for a doublecheck of passable vegetation detection in autonomous ground vehicles". In Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, pages 230–236, 2012.

[8] D.-V. Nguyen, Lars Kuhnert, and Klaus Dieter Kuhnert. "Spreading algorithm for efficient vegetation detection in cluttered outdoor environments". Robotics and Autonomous Systems, 60(12):1498 – 1507, 2012.

[9] D.-V. Nguyen, Lars Kuhnert, and Klaus Dieter Kuhnert. Structure overview of vegetation detection. "A novel approach for efficient vegetation detection using an active lighting system". Robotics and Autonomous Systems, 60(4):498 – 508, 2012.

[10] Zhuokun Pan, Fang Wang, Lihua Xia, and Xiaoxuan Wang. "Feature extraction for urban vegetation stress identification using hyperspectral remote sensing". In Information Science and Engineering (ICISE), 2010 2nd International Conference on, pages 250–254, 2010.

[11] B. Zafarifar and P.H.N. de With. "Grass field detection for TV picture quality enhancement". In Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on, pages 1–2, 2008.

[12] Gonzalez, R.C., R.E. Woods, S.L. Eddins, *Digital Image Processing Using MATLAB*, New Jersey, Prentice Hall, 2003, Chapter 11.

[13] Isabelle Guyon and André Elisseeff. 2003. "An introduction to variable and feature selection". *J. Mach. Learn. Res.* 3 (March 2003), 1157-1182.

[14] Japkowicz, N., Shah, M. "Evaluating Learning Algorithms: A Classification Perspective". Cambridge University Press, New York, NY, USA. 2011.

[15] Christopher J. C. Burges. 1998. "A Tutorial on Support Vector Machines for Pattern Recognition". *Data Min. Knowl. Discov.* 2, 2 (June 1998), 121-167. DOI=10.1023/A:1009715923555 http://dx.doi.org/10.1023/A:1009715923555.

[16] Robert V. Krejcie and Daryle W. Morgan. "Determining Sample Size for Research Activities". In Educational and Psychological Measurement, #30, pp. 607-610. 1970.

[17] Dougherty, E. "An introduction to morphological image processing". Tutorial texts in optical engineering. SPIE Optical Engineering Press. 1992.