

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224329241>

Visual tree detection for autonomous navigation in forest environment

Conference Paper · July 2008

DOI: 10.1109/IVS.2008.4621315 · Source: IEEE Xplore

CITATIONS

21

READS

432

3 authors, including:



Fredrik Georgsson

Umeå University

28 PUBLICATIONS 236 CITATIONS

[SEE PROFILE](#)



Thomas Hellström

Umeå University

76 PUBLICATIONS 670 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Mammographic image analysis [View project](#)



SAMMIR: Safe multi-modal interaction with robot manipulators [View project](#)

Visual Tree Detection for Autonomous Navigation in Forest Environment

Wajid Ali, Fredrik Georgsson and Thomas Hellström

Abstract—This paper describes a classification based tree detection method for autonomous navigation of forest vehicles in forest environment. Fusion of color, and texture cues has been used to segment the image into tree trunk and background objects. The segmentation of images into tree trunk and background objects is a challenging task due to high variations of illumination, effect of different color shades, non-homogeneous bark texture, shadows and foreshortening. To accomplish this, the approach has been to find the best combinations of color, and texture descriptors, and classification techniques. An additional task has been to estimate the distance between forest vehicle and the base of segmented trees using monocular vision. A simple heuristic distance measurement method is proposed that is based on pixel height and a reference width. The performance of various color and texture operators, and accuracy of classifiers has been evaluated using cross validation techniques.

I. INTRODUCTION

Autonomous navigation of vehicles in off-road environment has already got a great deal of attention in last few decades. One research project, the IFOR Navigation (*Autonomous Navigation for Forest Machines*) [1] is going on at the Department of Computing Science, Umeå University. One important part involves detection and avoidance of known or new obstacles appearing on the path and on both sides of vehicle (forwarder shown in Fig. 1). In some cases e.g. if a tree on the left and the right side of vehicle is too close, the system should stop the vehicle and alert the remote human operator, who should be given the option of manually correcting the vehicle position, or giving the system the green signal to go ahead along the original recorded path. This paper presents a computer vision system that detects tree obstacles on the left and the right side of an autonomous forest vehicle and estimates the distance between a forest vehicle and the base of detected trees in order to make the navigation safe.

The article is organized as follows: Section II describes the related work. Section III presents tree detection and distance estimation algorithm. Section IV provides the implementation details. In section V, we present the experimental results followed by discussion. Section VI has the conclusions and future works.

II. RELATED WORK

Extensive research work has already been done in range-based obstacle detection, and there is also much work has been done in range and reflectance based hybrid obstacle

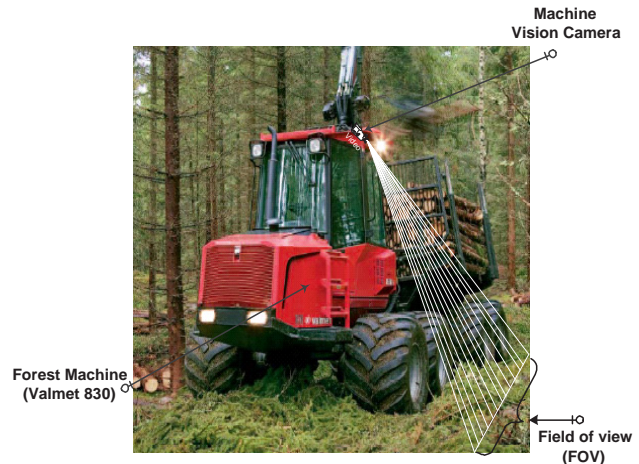


Fig. 1. Valmet 830 Forwarder, a forest vehicle

detection but considerable less attention is paid in only reflectance based object detection. Interestingly, the first autonomous mobile robot Shakey [15] had a simple appearance based obstacle detection in textureless background. Ulrich and Nourbakhsh [4] have also developed appearance based obstacle detection system that is based on passive color vision. Another similar system has been created by Batavia and Sanjiv. They used only color information for object detection [2]. Korah has used vision based texture cues for desert road following [11]. Another interesting system developed by Ollis and Stentz [8] navigated an industrial alfalfa harvester by following the cut/uncut crop line in a field. To do this, a color image was taken of a field, and a best fit step function was computed for each image scan line using the Fisher linear discriminant in RGB space. More information on appearance based related work can be found in [4]. A range-based obstacle detection system has been developed by Bostelman, Hong and Madhavan, they have used only TOF range camera for object detection towards AGV safety and advanced navigation [14]. Many researchers have also fabricated obstacle detection systems based on hybrid appearance and range information. In this category, a system developed by Christopher [12] can be used. He has used laser range, color and texture cues for autonomous road following. Similarly, Manduchi, Castano, Talukder and Matthies have successfully used hybrid techniques in obstacle detection and classification for autonomous navigation in an off-road environment [10]. Another related algorithm, though the purpose was somewhat else has been created in the same context as the one we have used in our project.

For example, Jeffrey and Sanjiv have created an image segmentation for forest inventory [3]. They have used texture cue for segmentation of images into tree and free space. The co-occurrence matrix has been employed to describe the texture of trees in images. However, the co-occurrence method is computationally expensive and thus can not be used in real time applications.

All in all, if we closely compare our system with other appearance based systems [11] then most of the developed systems have used only color or only texture. Color alone or texture alone can be expected to perform good in forest environment. Color only can not work well because of illumination problem and texture alone also can not give much information that can be used reliably in detecting the trees in variety of forest environment. However, texture can be utilized even in night vision and texture allows a robotic system to navigate autonomously [6], [9]. Thus we have used an integration of color and texture for tree detection in a forest environment.

III. TREE DETECTION AND DISTANCE MEASUREMENT

Generally, computer vision system do segmentation as a first step before feature extraction and classification process. Our tree detection method works in a reciprocal way like, first we do classification using computed features and then we perform segmentation based on classification results. We employ only non-geometric local image properties such as color and texture.

The **Tree Detection and Distance Measurement (TD&DM)** algorithm has three main parts: Training, Classification, and Segmentation and Distance Measurement.

A. Training

- Divide the training images (size of 320x240) into small non overlapping blocks (e.g. 15x15 pixels) and classify the blocks manually, for example blocks containing tree object. Tree has two classes: brown tree trunk and black tree trunk. Background has four classes: leaves, snow, bushes with snow, and leaves with snow. Example images from our data can be seen in Fig. 2.
- Perform some pre-processing operations on each block. Compute the features (e.g. color and texture) from each block using feature descriptor (e.g. color histograms and LBP - Local Binary Patterns [5]) and store them as a vector.
- Train the ANN/KNN using the extracted feature vectors.

B. Classification

- For each test image, perform some pre-processing operations (e.g. image smoothing using a gaussian filter).
- Divide each pre-processed image into small non overlapping blocks. Compute the features from each block and feed these features to the classifier.
- The classifier classifies all the blocks of the image based on extracted feature vectors. After the classification process, each block belongs to a particular class.



Fig. 2. Three data set of acquired images (Images in first, second and third row are belong to data set I, II and III respectively)

C. Segmentation and Distance Measurement

- Based on a class type, all blocks of the image are denoted either tree object or background. White color is assigned to tree objects and black color to background objects. After this step, we will have a binary image.
- Perform some post-processing operations to improve the segmentation (e.g. binary area open, morphological closing and filling). This results in a segmented image.
- After the segmentation step, each tree object is labeled. For each labeled tree object, the minimum row is found. The row number is used to calculate the distance between the tire and the tree object.
- The minimum distance to the tire is sent to control system of the vehicle for further decisions.

IV. IMPLEMENTATION

A. Features

Once the image is captured, the image is resized to 320x240 pixels in resolution. This step is done both to speed up the image processing and tree detection procedure. In experiments, we have observed that images with 320x240 size still preserve good enough color and texture information. We split the image into blocks after smoothing and color transformation steps. In experiments, different block sizes such as 8x8, 12x12, 15x15, 24x24 have been applied and tested. We have selected a block size of 15x15 for images having 320x240 pixels in resolution because this block size is the most suitable in terms of time complexity and recognition rate. In this case we have a total of 336 non-overlapping blocks in one image.

After splitting process, color and texture features are extracted using selected descriptors. Two color descriptors (i)- Mean & standard deviation (ii) Color histograms, and three texture descriptors (i) The co-occurrence matrix [13] (ii) Gabor filters [7] (iii) LBP - Local Binary Patterns are evaluated. In addition, four different color models (RGB, HSV, adapted Ohta, and L*a*b*) are examined. The complete list of feature descriptors, their different combinations

TABLE I

TOTAL NUMBER OF FEATURES BY COLOR AND TEXTURE ALONE (FROM ALL THREE CHANNELS OF AN IMAGE).

Descriptor Type	No. of Features (CA / TA)
MeanStd	6
ColorHist	45
LBP	768
LBPriu81	30
Multi LBP	162
Co-Occurrence	42
Gabor	144

TABLE II

TOTAL NUMBER OF FEATURES IN TWO DIFFERENT FUSION TECHNIQUES WITH VARIOUS COMBINATION OF COLOR AND TEXTURE DESCRIPTORS.

Descriptor Type	No. of Features	
	1 st Fusion Tech. CA + TG	2 nd Fusion Tech. CA + TA
MeanStd & LBP	262	774
MeanStd & LBPriu81	16	36
MeanStd & MultiLBP	60	168
MeanStd & Co-Occurrence	20	48
MeanStd & Gabor	54	150
ColorHist & LBP	301	813
ColorHist & LBPriu81	55	75
ColorHist & Multi LBP	99	207
ColorHist & Co-Occurrence	59	87
ColorHist & Gabor	93	189

and number of features extracted by respective descriptors are provided in Tables I and II.

We adopted a fusion of color and texture on feature vector level. We have used two further techniques to combine color and texture features. In the 1st feature fusion technique, first we compute color features from all three channels of an image and store these in three different vectors and then we construct a single feature vector by inserting all three computed color feature vectors in a sequence. Let us denote this single color feature vector as *CA*. The *CA* is then combined with gray-scale texture features *TG* (texture features are computed from gray-scale version of the same image). For the 2nd technique, we extract both texture and color features from all three channels of an image, the texture feature from all three channels is denoted by *TA*. In both tables MeanStd and ColorHist correspond to color features and the rest of the features correspond to texture features, MeanStd stands for mean intensity and standard deviation, and ColorHist stands for color histogram of an image.

B. Training of Classifiers

We have employed two classifiers: Artificial Neural Networks (ANN) and K-Nearest Neighbor (KNN). Artificial Neural Network has three layers including one hidden layer consisting of different hidden units are used that is based on the length of the feature vector. The weights are updated using gradient descent back-propagation. We have performed certain preprocessing steps to make the neural network training more efficient. We used an approach for scaling the network inputs and targets to normalize the mean and

standard deviation of the training set. The parameters for neural network training are dependent on the number of features as input to classifier, the number of outputs and the number of samples used for training. For example we have used 10 hidden nodes, 1000 epochs and 0.001 values as goal in case of joint *color histogram* and rotational invariant & uniform LBP (*LBPriu81*) texture features. In this particular scenario we have used 300 samples with 6 different classes: two classes for tree (brown tree trunk and black tree trunk), and four classes for background (leaf, snow, leaf with snow and bushes with snow). The 75 number of features are extracted from each sample as exhibited in Table II, after applying PCA on extracted data, we obtain 24 number of features. The number of hidden nodes has been calculated based on available sample data for training and their output classes using formula given in equation 1. The epoch and goal values have been fixed by iteratively analyzing the training level to meet the performance goal.

$$n_i n_h + n_h + n_h n_o + n_o = TOS \quad (1)$$

where n_i , n_o and n_h are the number of input nodes, output nodes, and hidden nodes respectively. *TOS* is the total number of free parameters in the model and should not exceed the number of samples.

The KNN classifier is not to be trained due to the nature of its design. We simply store the features in suitable form for classification by KNN. In the KNN ($k = 3$) classification, the distance is measured according to the selected feature descriptor, e.g. log likelihood distance measure and Mahalanobis distance are used for LBP and for the co-occurrence matrix respectively. In case of joint LBP and color histogram features, we have used a simple yet efficient histogram intersection as a distance measure in KNN classification. Euclidian distance is used with KNN for both Gabor filter and simple mean intensity and standard deviation of color features.

The classification plays a central role in tree detection system because tree segmentation entirely relies on the performance of classification. The preprocessing works in the same fashion like it did in the training part of the system. After division of the image into blocks, the features are extracted from each block starting from left to right and top to bottom in test image. Then these feature vectors are fed into selected classifier to classify the blocks of image as tree or background. After the classification step we obtain a binary image, white area represents tree and black area as background. The original image and classified binary image can be seen in Figure 3(a) and 3(b) respectively.

C. Post-processing and Distance Estimation

After classification we performed morphological post processing like binary area open, image closing, hole filling and boundary extraction in order to make the segmentation fine.

There are various computer vision techniques that can be used in estimating the distance between the camera (or tire of vehicle) and detected objects in scene image. In case of stereo vision, the camera must be fixed, calibrated

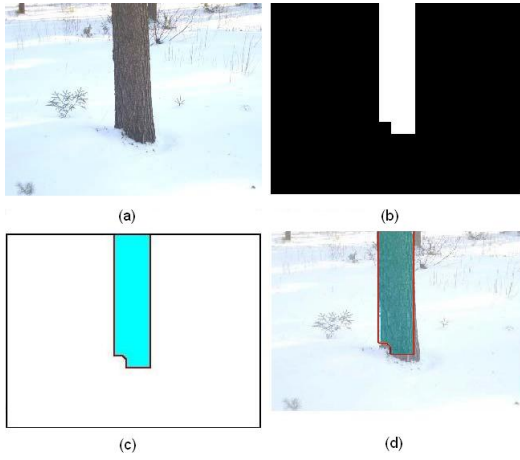


Fig. 3. Representation of classification and boundary detection. (a) is an original image, (b) is a binary classified image, (c) is a color labeled version and (d) is super-imposition of labeled image on original image

away distance from each other all the times, making a bulky system. The measurements by stereo vision systems can be inaccurate too due to possible parallax errors. The multi camera system has also many problems similar to stereo vision system. Monocular vision based techniques can also be employed to measure the distance in various ways. For example, we have formulated and derived mathematical equations for the purpose of distance estimation using few camera parameters and some other information like width of the tire and height of the camera. The formulas are given in equation 2. The camera setup and other values that we have used in estimating the distance can be seen in Fig. 4.

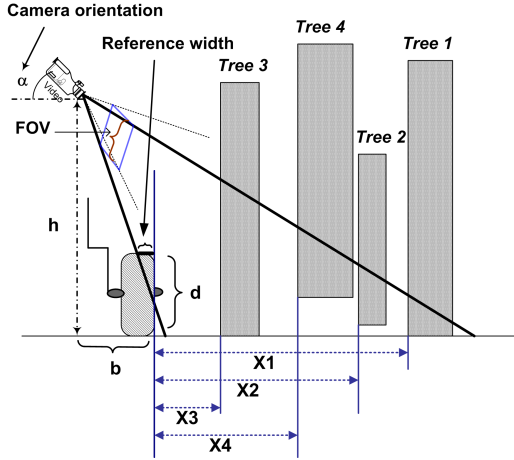


Fig. 4. Distance measurement geometry.

$$X = \frac{bd}{h} + \sqrt{\Delta \tan \alpha + \Delta^2} \quad (2)$$

where

$$\Delta = \sigma \sqrt{\frac{h^2 + (b+y)^2}{f^2 + \frac{\sigma^2}{4}}}$$

In the above formula, f is a focal length of the camera, α is the tilt/orientation of the camera, σ is the distance between

the projection of the tire and the base of tree and X is distance to the base of the tree. The above mentioned monocular vision based distance estimation method still needs some camera parameters and other values like height and orientation of the camera. This single camera based distance measurement system can also make wrong estimations due to constant environment and some other practical difficulties. Due to these hard problems, we employed a heuristic and simple approach to estimate the distance between the tire and the base of detected trees by assuming that the ground is relatively flat and there are no overhanging other objects in the scene. These assumptions are reasonable in variety of outdoor environments and if these two assumptions are valid then the distance can be computed between the base of tree and the tire of vehicle based on the concept of pixel ratio (pixel height) and the width of tire as a reference frame. Furthermore, this approach also works well if there are more than one trees in an image. The formula of this linear dependant approach is given in equation. 4.

$$\frac{X_A}{X_P} = \frac{L_R}{L_P} \quad (3)$$

$$X_A = \frac{L_R}{L_P} X_P \quad (4)$$

where X_A is an actual distance from tire to the base of tree, X_P is a pixel distance from tire to the base of tree in image, L_R is an actual reference width and L_P is a pixel width of tire. We are actually mapping the pixels in an image to the actual distance from the base of the tree.

D. Overall System Design

Figure 5 shows the simplified architecture of tree detection and distance measurement system. The system contains a CCD camera mounted on the forest machine and a mobile computer that executes the proposed monocular vision based system. The camera captures the tree images and provide these images to vision system for their analysis. Vision system first detects the trees in image and then distance measurement algorithm estimates the distance between recognized tree and the tire of forest vehicle. If the system finds the minimum distance less than proximity threshold T then it sends the command to the vehicle's control system. Then control system stops the forest vehicle and alerts the remote human operator to take a further decision. After decision by remote human operator, the vision starts again to analyze new images in loop.

V. RESULTS AND DISCUSSION

A. Experimental Results

We experimented with two classifiers to asses the performance of a number of color and texture features both independently and in combinations by using various kinds of suitable descriptors. We have evaluated all selected feature descriptors and classifiers on three image data sets I, II, and III that we collected for research purpose. Most of the presented results in this paper are obtained from

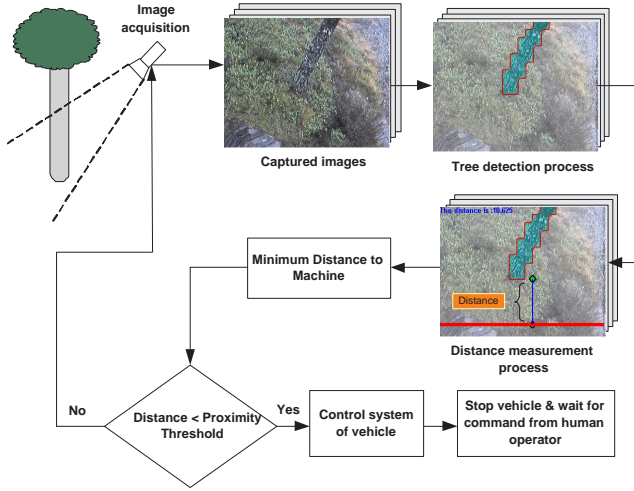


Fig. 5. Simplified architecture of tree detection system

image data set II. In addition to this, every experiment is repeated in each of four different color spaces (RGB, HSV, adapted Ohta and $L^*a^*b^*$). Since the accuracy of the tree detection/segmentation algorithm depends on the accuracy of the small image block's classification. We have estimated the performance as the misclassification rate of the classifier using cross validation techniques. In total 108 results are fetched in each of the two classification techniques, only the first 10 results are shown in the Tables III and IV. The results are based on mean percentage of False Negative (F.N) and False Positive (F.P) by each feature descriptor.

The performance results from our empirical study show that neither color nor texture alone can give the optimal performance. Integration of color and texture shows the optimal and satisfactory performance as shown in Tables III and IV. It is also clear that the fusion of color histograms with LBP and its extended versions like rotational invariant & uniform LBP (LBPriu81) and multi spectral LBP has usually performs better than other feature descriptors but in few cases color histogram with Gabor filter also performed well. It is well known that color always beats texture in static illumination conditions but texture is much more robust than color with respect to changes in illumination. In addition to this, texture can also be utilized with night vision [9]. However, the integration of color histogram with LBP (and its sub-types) has shown optimal performance in both exploited fusion techniques. The presented results in Tables III and IV show that the HSV has slightly out performed other color spaces in terms of illumination effects, but Ohta and RGB also show promising results. As far as comparison of classifiers is concerned, the KNN has performed better and has lower misclassification rate than ANN. The KNN is computationally more expensive than ANN which can be a disadvantage for real-time application.

B. Discussion

Tree detection and accurate distance measurement for navigating a forest vehicle in forest environment, is a dif-

TABLE III
CLASSIFICATION ERROR BY KNN USING K-FOLD CROSS VALIDATION METHOD. DATA IS SORTED BASED ON RESULTS BY TESTING SET.

Color Space	Fusion Tech.	Feature	Training Set [%]	Testing Set [%]
HSV	1 st	ColorHist & MultiLBP	4.899	5.3427
RGB	2 nd	ColorHist & LBP	9.1808	6.2735
Ohta	1 st	MeanStd & LBP	8.125	7.5
RGB	1 st	MeanStd & Gabor	8.5344	7.5862
RGB	2 nd	MeanStd & Gabor	10.473	7.594
RGB	1 st	MeanStd & Co-Occ	14.138	7.8447
RGB	2 nd	MeanStd & Co-Occ	13.143	7.8759
Ohta	1 st	ColorHist & Gabor	8.0201	8.2389
HSV	1 st	ColorHist & LBPriu81	5.8461	8.8708
RGB	2 nd	MeanStd & MultiLBP	7.571	9.0038

TABLE IV
CLASSIFICATION ERROR BY ANN USING K-FOLD CROSS VALIDATION METHOD. DATA IS SORTED BASED ON RESULTS BY TEST DATA SET.

Color Space	Fusion Tech.	Feature	Training Set [%]	Testing Set [%]
Ohta	2 nd	LBPriu81	28.771	12.152
HSV	2 nd	ColorHist & CoOcc	22.17	12.941
RGB	1 st	ColorHist & LBPriu81	32.155	13.224
RGB	2 nd	ColorHist & LBPriu81	19.093	13.502
HSV	2 nd	ColorHist & Gabor	7.2694	13.676
HSV	1 st	ColorHist & LBPriu81	9.2743	13.709
RGB	1 st	ColorHist & LBP	41.12	15.397
HSV	2 nd	ColorHist & LBPriu81	7.977	15.699
RGB	1 st	ColorHist & Gabor	12.6	16.061
Ohta	2 nd	ColorHistogram	24.229	16.302

icult and very challenging task. The described segmentation method works well even for images with a wide variety of illumination, non-homogeneity in bark texture, shadows, and different shades of color in even a single tree object. The effect of foreshortening and non homogenous appearance of objects set extra requirements. Due to these high variations the conventional computer vision algorithms such as threshold or edge based segmentation might not work at all. Thus, we have developed a new segmentation method that can handle the problems and situations that we have discussed above. However, it is clear that, the result of tree detection is greatly affected by the performance of small block's classification. Some-times the system does not detect trees that are far away in view or due to other causes that we have discussed in the previous section.

It might be possible that the manual selection of small samples for classifier training could cause problems. It is also clear that the usage of small image blocks instead of pixels could create problems in the segmentation of tree that is purely rely on classification of such small image blocks that preserve pixels from more than one class. This problem could be solved by using SOM based visual training and classification [6]. We have used a simple and heuristic technique to estimate the distance between the tire of forest vehicle and the base of the tree instead of using a method based on camera calibration. The method is based on pixel ratio and the width of the tire as a reference in an image. In this way the system does not rely on camera mounting pose

and other camera parameters.

The presented experiments has two key aims. Firstly, it investigates whether the developed monocular vision based tree detection and distance measurement method can be used reliably or not? Secondly, these experiments allow a comprehensive comparison between selected feature descriptors in different combinations, the effect of using a range of various color spaces for feature extraction, and last but not least the accuracy comparison of selected classifiers (ANN and KNN).

The results presented in the last section clearly indicate that the combination of color information and texture features increases the accuracy of the tree segmentation process that is purely based on block classification. The empirical study shows that the color histogram and multi spectral LBP feature fusion approach has a classification accuracy of 94.7% followed by the fusion of MeanStd and Gabor filter features (92.4% accuracy), and co-occurrence with MeanStd features (92.1% accuracy). This accuracy is based on classification by KNN while evaluating the performance using K-Fold cross validation. If we compare only color features with each other, color histogram performs better than mean intensity & standard deviation of color. The comparison of texture operators shows that LBP and its sub-types have higher accuracy due to its tolerance against illumination compared to Gabor filter and co-occurrence matrix. Concerning the processing time, the LBP proved to be the fastest texture operator due to its computational simplicity. The computational complexity associated with other texture feature extraction techniques is much higher than LBP since the calculation of co-occurrence matrices are computationally expensive and the Gabor filter takes much more execution time due its bigger kernel size.

The comparison of classifier's accuracy shows that the KNN is more efficient than ANN. A possible reason for the bad performance of ANN could be that we have a fixed number of hidden nodes for all type of features. A better approach could be to adjust the number of hidden nodes based on the length of feature vector.

The fusion of LBP and color histogram features can be reliably employed for tree detection task and have performed well in terms of both accuracy and execution speed. Hence, the best result is 94.7% that has been obtained by the combination of LBP and color histograms, in the first fusion technique, using HSV color space and KNN classification method.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

Classification based tree detection and distance estimation method for autonomous navigation in forest environment is presented in this research work. Color and texture cues were combined to yield better performance than individual cues could accomplish. The fusion of LBP and color histogram features have shown an excellent performance. A simple heuristic distance measurement method has been used to estimate the distance between the base of the tree and the tire of vehicle based on pixel ratio and the width of tire as a

reference. The segmentation and distance estimation procedure described here assumes that the ground is relatively flat and there are no overhanging other objects in scene image. If these two assumption are valid then the proposed method works fairly well in estimating the distance.

B. Future Works

The KNN is a good classifier but it is computationally more expensive and cannot be used for real time applications. SOM and Support Vector Machines can increase the classification accuracy and can speed up the tree detection process. Matlab is not meant to develop any real time system, thus development of tree detection in C++ language can make this system real time. Best results can be achieved by adding few features of TOF 3D range camera if it has a true range up to at least 10 meters. Initial tests show how the developed method for distance estimation looks promising. More detailed experiments regarding this is subject to future work.

REFERENCES

- [1] T. Hellström, T. Johansson and O. Ringdahl. *Development of an Autonomous Forest Machine for Path Tracking*, Volume 25 of *Springer Tracts in Advanced Robotics*, pages 603-614. Springer, Field and Service Robotics, 2006.
- [2] Parag H. Batavia and Sanjiv Singh. Obstacle detection using adaptive color segmentation and color stereo homography. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.
- [3] Jeffrey Byrne and Sanjiv Singh. Precise image segmentation for forest inventory. Technical Report CMU-RI-TR-98-14, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1998.
- [4] Ulrich I. and Nourbakhsh I. Appearance-based obstacle detection with monocular color vision. *AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000.
- [5] Topi M. and Pietikäinen M. Texture analysis with local binary patterns. *World Scientific*, 2005.
- [6] Turtinen M. and Pietikäinen M. Visual training and classification of textured scene images. *Proc. The 3rd International Workshop on Texture Analysis and Synthesis*, pages 101-106, 2003.
- [7] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837-842, 1996.
- [8] M. Ollis and A. Stentz. Vision-based perception for an automated harvester. *Proceedings of IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS 97)*, 1997.
- [9] R. Manduchi R. Castano and J. Fox. Classification experiments on real-world texture. *Third Workshop on Empirical Evaluation Methods in Computer Vision*, pages 0320, 2001.
- [10] A. Talukder R. Manduchi, A. Castano and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Springer Science and Business Media, Inc., The Netherlands*, 2005.
- [11] C. Rasmussen and T. Korah. On-vehicle and aerial texture analysis for visionbased desert road following. *IEEE International Workshop on Machine Vision for Intelligent Vehicles, MVIV-05, San Diego.*, 2005.
- [12] Christopher Rasmussen. Combining laser range, color, and texture cues for autonomous road following. *In Proc. IEEE Inter. Conf. on Robotics and Automation*, Washington, DC, 2005.
- [13] K. Shanmugam R.M. Haralick and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610-621, 1973.
- [14] T.H. Hong R.V. Bostelman and R. Madhavan. Towards agv safety and navigation advancement-obstacle detection using tof range camera. *International Conf. On Advanced Robotics(ICAR), USA*, 2005.
- [15] N.J. Nilsson. Shakey the robot. technical note 323. *SRI International.*, 1984.