# Towards Integrated Image Contrast Models in Segmentation of Trees

Gabriel da Silva Vieira[*§1], Fabrizzio A.A.M.N. Soares[*2], Samuel A. Santos[§3], Gustavo T. Laureano[*4], Junio Cesar de Lima[§5], Ronaldo M. Costa[*6], Juliana Paula Félix[*7], Thamer H. Nascimento[§8]
[*]Federal University of Goiás, *Pixelab Laboratory*. Goiânia - GO, Brazil.
{fabrizzio[2], gustavo[4], ronaldocosta[6], julianafelix[7]}@inf.ufg.br
[§]Federal Institute Goiano, *Computer Vision Laboratory*. Urutaí - GO, Brazil.
{gabriel.vieira[1], junio.lima[5], thamer.nascimento[8]}@ifgoiano.edu.br, samuelalvesv4[3]@gmail.com

***Abstract*—Computer vision is an area in high demand which is bringing new trends for urban and rural applications. Some examples can be found in autonomous navigation projects, monitoring services, fruits/grain harvesting, pest control, and so forth. However, drastic or even unperceptive changes in the image acquisition process limit the development of these applications, especially for problems that require solutions for uncontrolled environments such as outdoor areas. Thus, the definition of what a machine is looking at is a challenging task. In this study, we dealt with the image segmentation problem in order to develop a method to delineate tree trunks, their branches, and foliage. As tree detection is a crucial topic in mobile robotics, we investigated it to give an initial interpretation of external scenes. We prepared an image dataset to validate the proposal in which two classes were defined, tree and non-tree. The pixels of each image were classified based on the proposed method, and the results show that our method obtained a positive result of $91\%$ accuracy.**

**Keywords:** image segmentation; tree trunk detection; feature engineering; computer vision.

## I. Introduction

Computer vision has been widely used in science, industry, and entertainment. It is an area in high demand which is fostering the progress of computer science but also other areas of research. Usually, it is a piece in the complex process of automation, medical imaging processing and analysis, autonomous navigation, services of ostensive monitoring, urban mobility policy, and immersive games.

A common challenging into computer vision area consists of the definition of what a machine is looking at. It means that the computer needs to be prepared to understand the world around it. A significant number of techniques can be used to deal with this task, and typically a lot of them are combined to achieve the desired results.

In this way, input data are preliminarily treated to characterize objects or a portion of them. It is a well-known step termed feature engineering, which is responsible for preparing the computer to recognize and classify data. This procedure plays a crucial role in vision applications and therefore is always treated with care.

When it comes to external scenes, the extraction of key features can be quite complex due to the exposure of various adverse conditions, such as frequent changes in illumination that creates artifacts like shading, photometric distortion, and noise. Because of that, image contrast is a recurring problem in digital image processing that can increase the challenge in developing computer-based solutions, especially in scenes that have low contrast settings.

Image segmentation plays a key role in providing image data to be analyzed, which influences overall success in understanding the image [1]. Its general proposal is to differentiate the object of interest from its background by following some kind of pixel characteristic as intensity, texture, color, or gray level information. For example, intensity-based methods are concerned with setting thresholds, discontinuity-based methods consider intensity variation among neighboring pixels, and region-based methods deal with the definition of homogeneity criteria.

For any type of approach, image contrast can be used. When different objects are in contrasting areas, a segmentation method can be well-performed to label them; thus, this differentiation can be used to define thresholds, discontinuities, and homogeneous regions, which provides a starting point to segmentation approaches as well as to object detection and delineation methods.

In this study, we explore the contrast in outdoor environments to detect the trunk and foliage of trees. In general, this is one of the main requirements of mobile robotics because it gives orientation to the autonomous movement, the position of the object in the surrounding scene and safe navigation to avoid prominent obstacles. For this reason, tree detection is being investigated and used as part of intelligence systems in fruit harvesting and, it going beyond, it has the potential to be applied in the monitoring of urban areas, as public squares.

Considering these notes, the remainder of the paper is organized as follows. After briefly reviewing closely related work in Section II, we show an overview of the proposed technique which describes our segment trees algorithm in Section III. The experimental results and analyses are given in Section IV and Section V concludes the paper.

## II. Related Work

To deal with the tree detection task, Mendes et al. [2] proposed a vision-based detector which uses the local binary pattern codes (LBP) for textured image description. First, they used the Sobel operator to detect key-points of vineyards. After that, the LBP descriptor was applied to extract portions of trees around them. Tests showed that their best result was achieved with the use of an RGB camera.

Juman et al. [3] presented a trunk detector method that uses color space combination to enhance the contrast between the ground and other objects. They observed that the best contrast was obtained when *hue* value of the HSV color space was subtracted from the *blue* channel from RGB. It was one of the steps in their method to develop a mobile robot for data collection and navigation in an oil-palm plantation.

Shao et al. [4] proposed a method of recognizing tree trunks based on Hough Transforms of the L*a*b* color space. Their method was used to segment and detect tree trunks in outdoor fields under different illumination conditions by using an RGB camera.

Lu and Rasmussen [5] developed a contrast-based approach to detect trees in outdoor areas. They considered that tree trunks are different in appearance from their background, i.e., in an opposite contrast. This note was used to design a kernel function to extract the features of interest.

Yıldız [6] considered trees as persistent visual landmark features in outdoor settings. Hence, they developed a method that incorporated color and regional attributes in the detection of tree trunks. Besides, they prepared a dataset with outdoor images of urban areas that contained one or more trees in a variety of poses in the RGB color space.

In the same way, Teng et al. [7] considered complex scenes from urban areas to propose a method to identify the trunk structure of trees and also the identification of leaf regions. Other researchers are using unmanned aerial vehicles to detect individual trees in residential and rural areas through their crowns [8], [9], and others are looking at fruit harvesting using machine vision to detect citrus fruit and tree trunks [10].

To contribute to this area, we present a new method for segmenting trees from uncontrolled environments. Similar to previous approaches, we deal with the challenge of detecting trees, but we propose advancements by delineating tree regions with the preservation of trunks, branches, and foliage. In general terms, we focus not only on trunk detection but also on tree segmentation. Thus, the proposal has the potential to be used as a step in machine vision approaches as well as to be used as a feature engineering strategy in the training and classification of outdoor scenes.

## III. Proposed Method

The proposed method starts by taking into account an RGB image as input. This image, $I$ is cut into small slices in which the next slice overlaps a part of the previous one. We used this strategy to be more accurate in detecting branches and trees that are further from the camera and also to deal with bright artifacts close to the trees. In the tests, the size of the slices was defined as 20 rows $\times$ $d$ columns, where $d$ was the width of $I$.

Considering that natural light produces a relevant contrast between the background and foreground of a scene, we used the kernel function proposed by [5] to create models that extract vertical features of trees. Eq. 1 reproduces this contrast model function.

$$x' = y\sin(\theta), y' = y\cos(\theta)$$
$$B(x,y) = \exp(-0.5 \cdot (\frac{x'^2}{s_x} + \frac{y'^2}{s_y}))\cos(2\pi f x'), \tag{1}$$

where $s_x$ and $s_y$ are half width and height of the kernel, $x \in [-s_x, s_x], y \in [-s_y, s_y]$. The frequency and orientation of the model filter are set to be $f = 1/(2 * s_x)$ and $\theta = 90°$, respectively.

These models are convolved with the input image $I$ to produce curves of contrasting areas: $F = (B * I)$. Each model filter produces its curve through the summed-up value of each column of $F$, Eq. 2.

$$curve(1,j) = \sum_{j=1}^{n}\sum_{i=1}^{m} F(i,j) \tag{2}$$

$curve$ is a vector containing the sum of the values of all rows $i$ of each column $j$ of the $F$, $n$ is the number of columns and $m$ is the number of rows of $F$.

The valleys of the curves indicate dark areas that contrast with a bright background. Thus, the width of these valleys is measured and based on them some bounding boxes, or patches, are placed in the slice of the image $I$ to select only those areas.

As the best model filter $B$ is unknown, it is necessary to explore different filter sizes. Consequently, each one of them forms their own bounding boxes and to select the most appropriate for an area we used the Greedy Non-Maximum Suppression (GreedyNMS) [11].

To reduce the influence of brighter points inside the patches, a local evaluation is applied to discard these points. These patches are converted to the HSV (Hue, Saturation, Value) color space and the channel *value* $v$ is used to detect points with high values, (Eq. 3).

$$\overline{v} = \frac{1}{n}\sum_{i=1}^{n} v_i \qquad \sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(v_i - \overline{v})^2} \tag{3}$$

where $n$ is the number of pixels in a bounding box, and $i$ is the index of each one of them. Hence, $v$ passes through a threshold evaluation such that

$$V(i) = \begin{cases} 1 & \text{if } v_i < (\overline{v} + 3\sigma), \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The Eq. 4 is a logical binarization that is based on a condition which detect points further from the mean, $\overline{v}$, in 3 times the standard deviation, $\sigma$. Bright points inside a patch are labeled with 0 and other points are labeled with 1. Based on this result, points that were labeled with 0 are removed from the original patch (RGB image), and the others are maintained. After these steps, a partial segmented image, $I_s$, is obtained.

Until this moment, some of the non-tree points were discarded using local evaluations. However, these first steps are not enough to deal with the complexity of the outdoor scenes. Thus, a complementary approach is required.

A global evaluation is applied in the original image, $I$, as well as in the remained image, $I_s$. In the first case, a bright detection function was developed to detach trees from the background (bright points). In the second case, variability analysis, morphological operators and a tree attribute are used to detach trees from the ground.

A simple measure of variability is used to remove non-tree points. The mean absolute deviation (MAD) is employed considering the remaining points of $I_s$. Eq. 5 and 6 are responsible for calculating the MAD around the mean values, $\overline{V}$ and $\overline{U}$.

$$m' = \frac{1}{n}\sum_{i=1}^{n} |V_i - \overline{V}| \quad (5) \quad m'' = \frac{1}{n}\sum_{i=1}^{n} |U_i - \overline{U}| \quad (6)$$

where $V$ is the *value* channel from $I_s$ (converted to the HSV color space) and $U$ is calculated according to Eq. 7 (an intuitive equation developed by [12]).

$$U = G \cdot \max\left[(G - R), 0\right] \cdot \max\left[(G - B), 0\right] \qquad (7)$$

$R$, $G$ and $B$ are the three channels of $I_s$, red, green and blue, respectively.

Hence the MAD values, $m'$ and $m''$, are used to define a threshold that discards non-tree pixels. Eq. 8 checks if points of $I_s$ are in accordance with it. Points of $I_s$ at index $i$ that receive the value 0 are then discarded.

$$I_s(i) = \begin{cases} 1 & \text{if } (|V_i - \overline{V}| < th_1) \text{ or } (|U_i - \overline{U}| < th_2), \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

where $th_1 = \lambda_1 m'$ and $th_2 = \lambda_2 m''$, if the values of $\lambda_1$ and $\lambda_2$ increase, then it will accept more bright and green points, respectively. In the experimental tests, we set them with the value 2.

In the same way, an energy function is considered to label bright points from image $I$. Firstly, the three RGB channels of $I$ are used, and a differential evaluation among them are employed to emphasizes areas of brightness. In Eq. 9 a weight is calculated for each pixel from $I$ and stored in $W$. After that, a logical validation is performed and a logical image, $T$, is yielded, Eq. 10.

$$W = B \cdot \max\left[(B - R), 0\right] \cdot \max\left[(B - G), 0\right] \cdot \max\left[(G - R), 0\right] \qquad (9)$$

To complete this step, we consider the HSV color space. The input image $I$ is converted to HSV and the logical image $T$ passes through an evaluation in which Eq. 11 checks the influence of bright points one more time to guarantee that only intense bright points will be discarded. Then, based on $T$, the image $I_s$ is updated to remove those detected points.

$$T(i) = \begin{cases} 1 & \text{if } W_i = 0, \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

$$T(i) = \begin{cases} 1 & \text{if } (T_i \neq [H_i > S_i]) \text{ and } (T_i \neq [H_i > V_i]), \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

where $H$, $S$ and $V$ are the channels hue, saturation and value from the converted image $I$ into the HSV color space.

Furthermore, as it can be noticed, trees have one common feature that qualifies them by their structures: trunks have a vertical, or a semi-vertical shape. We used this attribute to track tree trunks to ground contact. A user parameter, $0 < \gamma \leq 1$, controls the portion of the image, $I_s$, that is used in this step. For example, if $\gamma = 0.3$ then in the first 30% rows of $I_s$ no changes is applied and in the others rows, an opening operation is employed to remove some small artifacts from $I_s$. After that, it checks the connection of the remaining points in the vertical direction with the borderline defined by $\gamma$. In the tests, $\gamma$ was set with the following values: 0.5, 0.7 or 0.8.

## IV. Experimental Results and Analyses

To evaluate the proposed method, we prepared a small dataset with 20 images These images were manually segmented, and they served as ground truth maps. It was made to measure the similarity between the results of the proposed method concerning these reference images. To the best of our knowledge, a proposal like that, i.e. a non-supervised segmentation method to this specific task is not easy to find, making it difficult to compare similar methods. Because of that, the presented results only refer to the proposed method, and the image dataset is available on-line[1] to stimulate new approaches.

To quantify the results, we used statistical measures to analyze the reliability of the statistical relationships. The problem was modeled as a binary classification test in which the samples were labeled as tree or non-tree. The classification assigned as correct or incorrect was defined as:

- True Positive: tree pixels correctly identified as tree.
- True Negative: non-tree pixels correctly identified as non-tree.
- False Positive: non-tree pixels incorrectly identified as tree.
- False Negative: tree pixels incorrectly identified as non-tree.

Based on this binary evaluation, we calculated some statistical rates. For example, the *sensitive* (recall, hit rate, or true positive rate - TPR) was used to measure the proportion of pixels which test positive for tree among those which were manually labeled as tree, and the *false positive rate* (FPR) which showed the proportion of non-tree pixels that were wrongly classified as a tree. Moreover, we used other ratios: true negative rate (TNR), positive predictive value (PPV), negative predictive value (NPV), false negative rate (FNR), false discovery rate (FDR), false omission rate (FOR), accuracy (ACC), and $F_1$ score.

Table I presents a list of statistical metrics that were modeled as a logical grouping of two sets, where $A$ is the ground truth image and $B$ is the output of the proposed method. In these equations, the function $n$ calculates the number of elements remaining in the set after a logical evaluation. For example, the True Positive Rate (TPR) is the number of elements in the intersection between $A$ and $B$ divided by the number of elements in $A$.

Table II presents the results obtained when these two classes are considered in each statistical rate. TPR and TNR achieved assertiveness results higher than 70%; the high value of PPV, 89%, told us how many of test positives are really true positives; and the NPV, in a smaller percentage compared to PPV, showed us how many of test negatives are true negatives. FDR

[1]The image dataset is available on https://github.com/vicom-ifgo-urutai/datasets/blob/master/dataset_001.rar

TABLE I
STATISTICAL METRICS.

| | |
|---|---|
| $TPR = \frac{n(A \cap B)}{n(A)}$ | $TNR = \frac{n(\neg A \cap \neg B)}{n(\neg A)}$ |
| $PPV = \frac{n(A \cap B)}{n(B)}$ | $NPV = \frac{n(\neg A \cap \neg B)}{n(\neg B)}$ |
| $FNR = \frac{n(A - (A \cap B))}{n(A)}$ | $FPR = \frac{n(B - (B \cap A)}{n(\neg A)}$ |
| $FDR = \frac{n(\neg A \cap B)}{n(B)}$ | $FOR = \frac{n(A \cap \neg B)}{n(\neg B)}$ |
| $ACC = \frac{n(A \cap B) + n(\neg A \cap \neg B)}{n(A) + n(\neg A)}$ | $F_1 = 2 * \frac{(PPV * TPR)}{(PPV + TPR)}$ |

and FOR complemented these results. In addition, the FNR reported only a margin of error of $5\%$ in data classification; ACC showed the proportion of correct results among the total number of pixels examined, in which the accuracy of the method was $91\%$; finally, $F_1$ tested the accuracy rate reporting a significance of $82\%$.

Fig. 1 shows the visual results of the proposed method. In the first column is presented the input images, and in the second column is presented the ground truth images that were segmented by hand. The third column presents the results of this proposal, and the fourth column shows the points that were wrongly classified, false positive and false negative. Looking at the outputs, we can note that the shape of the trees, branches, and part of the foliage have been preserved. Besides, the ground and the bright points in the background were substantially removed to maintain only the tree structures such that even the trees furthest from the camera were also preserved.

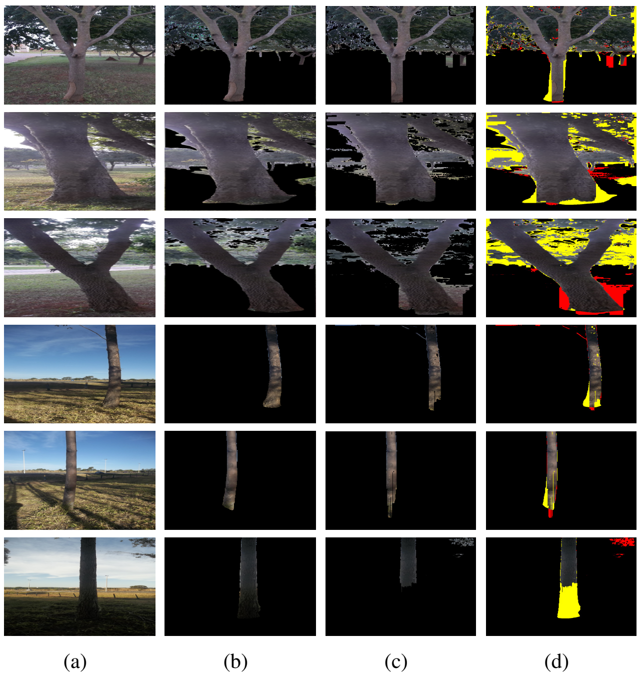

Fig. 1. Results from the proposed method. The input images $I$ are shown in column (a) and the ground truth in column (b). Column (c) shows the segmented outputs and column (d) shows the points at which the method did not hit, false positive (red) and false negative (yellow).

TABLE II
STATISTICAL MEASURES.

| TPR | TNR | PPV | NPV | FNR | FPR | FDR | FOR | ACC | $F_1$ |
|---|---|---|---|---|---|---|---|---|---|
| 0.77 | 0.93 | 0.89 | 0.87 | 0.22 | 0.05 | 0.10 | 0.11 | 0.91 | 0.82 |

## V. CONCLUSION AND FUTURE WORK

In this study, we investigated the image segmentation problem with particular attention to outdoor environments. We noted that the color spaces RGB and HSV could be used together to assist in this task. Moreover, contrast models are equally important because they can detect points of color in opposite directions; hence they can be used to distinguish points in juxtaposition or even in close association.

Considering these topics, we prepared a segmentation method and tested it in complex scenes from a public area. As the tree detection is an essential assignment in autonomous navigation, we explored one of the first challenges: the definition of what a machine is looking at. In our tests, the proposed method achieved $91\%$ of accuracy in detection tree structures, branches, and foliage.

In the next phase of the work, we intend to keep developing this method in order to label ground points and to distinguish tree trunks from their leaves. Furthermore, we want to apply this method in the extraction of features and use them in classification methods in order to increase the assertiveness of the classification.

## REFERENCES

[1] A. Khan and R. Srisha, “Image segmentation methods: A comparative study,” *International Journal of Soft Computing and Engineering*, pp. 2231–2307, 09 2013.

[2] J. Mendes, F. N. dos Santos, N. Ferraz, P. Couto, and R. Morais, “Vine trunk detector for a reliable robot localization system,” in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2016, pp. 1–6.

[3] M. A. Juman, Y. W. Wong, R. K. Rajkumar, and L. J. Goh, “A novel tree trunk detection method for oil-palm plantation navigation,” *Computers and Electronics in Agriculture*, vol. 128, pp. 172–180, 2016.

[4] L. Shao, X. Chen, B. Milne, and P. Guo, “A novel tree trunk recognition approach for forestry harvesting robot,” in *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*. IEEE, 2014, pp. 862–866.

[5] Y. Lu and C. Rasmussen, “Tree trunk detection using contrast templates,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1253–1256.

[6] T. Yıldız, “Detection of tree trunks as visual landmarks in outdoor environments,” Ph.D. dissertation, bilkent university, 2010.

[7] C.-H. Teng, Y.-S. Chen, and W.-H. Hsu, “Tree segmentation from an image.” in *MVA*, 2005, pp. 59–63.

[8] Y. Lin, M. Jiang, Y. Yao, L. Zhang, and J. Lin, “Use of uav oblique imaging for the detection of individual trees in residential environments,” *Urban forestry & urban greening*, vol. 14, no. 2, pp. 404–412, 2015.

[9] S. Selim, N. K. Sonmez, M. Coslu, and I. Onur, “Semi-automatic tree detection from images of unmanned aerial vehicle using object-based image analysis method,” *Journal of the Indian Society of Remote Sensing*, pp. 1–8, 2018.

[10] T.-H. Liu, R. Ehsani, A. Toudeshki, X.-J. Zou, and H.-J. Wang, “Detection of citrus fruit and tree trunks in natural environments using a multi-elliptical boundary model,” *Computers in Industry*, vol. 99, pp. 9–16, 2018.

[11] J. H. Hosang, R. Benenson, and B. Schiele, “Learning non-maximum suppression.” in *CVPR*, 2017, pp. 6469–6477.

[12] S. Eddins, “It ain’t easy seeing green (unless you have matlab),” 2014, [Accessed 20 Jul. 2018]. [Online]. Available: https://goo.gl/3ybj9q