Trunk Detection and Tree Disparity Calculation in Uncontrolled Environments

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

***Abstract*—Computer vision is an area have proven to play an essential role in urban and rural applications like medical, agriculture, and remote sensing. The use of image processing methods for simulating the visual capability of robots plays a crucial role in the consolidation of smart farming. The understanding of the complexity of outdoor environments, where the robot performs its task, is an essential issue for the development of efficient processes of autonomous mobility, especially in areas with uneven illumination, unpredictable weather conditions, and different color shades. In this study, we present a new method to detect and segment tree trunks from unstructured environments where natural properties such as lighting and terrain shape form a variety of non-controlled conditions. We prepared a dataset with stereo image pairs and ground truth maps to calculate disparities and to evaluate the proposed method in the application of smart farming. The results show that the presented approach can segment trees with high precision, which is an important step in calculating the disparity of external components by systems that use the stereoscopic view.**

***Index Terms*—stereo vision; image segmentation; disparity map; tree detection; smart farming.**

I. Introduction

Computer vision systems deal with such a complex amount of requirements that it is difficult to prepare a suitable computational model. Depending on the nature of the problem, not all operational constraints can be defined in such a way that the volatility in requirements increases the complexity in the design and construction processes. In such cases, statistical analysis, empirical evaluation, or even expertise are used to set at least some scenarios with controlled operating conditions to stabilize the requirements, e.g., in indoor environments such as a room, where the structural layout is typically known.

When it comes to outdoor environments some drawbacks are presented to the visual image analysis in which it has to face with sudden changes in scene lighting, unpredictable weather conditions, and with the effect of different color shades. Such circumstances require efficient techniques capable of self-adaptation to get optimum objects recognition, to simultaneously segment, classify and annotate data, and to recover the geometric properties of a scene.

In rural or semi-rural outdoor environments there is an increase in complexity. For example, the use of an autonomous robot in smart farming requires an accurate route during the working process, e.g., autonomous navigation through the plantations for the harvesting of fruits in smart farming. This navigation is, however, difficult to achieve due to regularly occurring natural obstacles such as tree during harvest. An autonomous robot needs to react to suddenly appearing tree, which may be even more complicated due to changes in the weather conditions, for example.

Although trees are outstanding elements of outdoor scenery, they vary in a substantial way, in shape, color, texture, intensity, and density of their branches. Besides, natural conditionals such as weather and lighting, and surrounding and background objects can affect the appearance of a tree due to shadows, brightness, and occlusions. Because of that, the recognition of trees or trunk detection is a challenging task which often uses a segmentation approach.

In general terms, computer vision systems apply a segmentation process as a first step before conducting image analysis whether in feature extraction, reconstruction, and in classification proposals for finding and identifying objects in an image or video sequence. The segmentation step allows dividing the complexity of a scene into small pieces of data to be handled in an isolated fashion. This strategy of reducing the complexity of a problem is a well-known design principle termed divide-and-conquer which is enforced in vision applications due to the various possible interactions, the diversity of inputs, and the expected results.

In outdoor robot navigation, if a tree is adequately detected, then it can be both a potential landmark as well as a prominent obstacle. The first one gives a reference point to the imaging system, whereas the second one alerts an area to be avoided. The tree detection is a valuable knowledge which is used to define autonomous path planning, to label a safe path and to specify the traversability of an area. To achieve the required result in robot navigation, the use of specific techniques accomplish tree detection in which the robot can reconstruct the scene by calculating the depth.

In addition, stereo vision methods can provide the necessary information to estimate the depth of objects in a scene. They are based on the stereoscopic imaging in which two or more images are collected from the same scene in different points of view. The projection of a point is traced in a pair of images, and its disparity is used in conjunction with camera

978-1-7281-2999-0/19/$31.00 ©2019 IEEE

parameters to measure the depth of this point. When it is done for all points, we can build a disparity map that holds the disparities between corresponding points in different image planes. Therefore, scene reconstruction can be performed by providing autonomy to the robot.

In smart farming applications, trunk detection is essential in harvests or pruning of trees by robots. A method that segments the trunk from its background and preserves contours and shapes is necessary to ensure the tree design. A harvesting robot demands precise information to localize and to detach the fruit, vegetable or grain, and to not damage the remaining crop. Likewise, precision information is required for estimating diameters and tree heights by computer vision systems deployed in the robot to be able to perform the prune a tree.

In this study, we present a new method to detect and segment tree trunks from unstructured environments where natural properties such as lighting and terrain shape form a variety of non-controlled conditions. It is based on the method proposed by [1] in which contrast templates were used to detect tree trunks. We prepared a dataset with stereo image pairs and ground truth maps to calculate disparities and to evaluate the proposed method in the application of precision agriculture.

This work is organized as follows. Section II shows published works related to our proposal. Section III shows our method of trunk detection and segmentation. Section IV shows the experimental results in which we evaluate the proposed method and its use in calculating disparities. Section V presents the conclusion and discussion of this paper.

II. Related Work

Tree detection has been an active area of research in off-road mobile navigation as well as in traversability classification and mapping, fruit localization, and forest inventory. Currently, digital image processing and machine learning techniques are applied to develop computer vision-based systems in which trunk, branches, and tree foliage detection are used as natural landmarks [2], to define safe crossing areas [3], and to apply automatic dendrometry [4].

As outdoor environments have a variety of adversities, most of the autonomous navigation proposals for rural or semi-rural areas treat the problem of detecting trees as a machine learning assignment. Thus, samples of trees and non-trees are prepared with a key-feature extractor [3], [5], with a self-learning framework [6], or even by hand [7], [8].

Ali et al. [7] employed two nor-geometric local image properties, color, and texture, in their tree detection method for autonomous navigation. They adopted a heuristic distance measurement to estimate the distance between forest vehicle and the base of segmented trees using monocular vision. In conclusion, they noted that neither color nor texture alone can give the optimal performance and that HSV and RGB color space can provide promising results.

Under the assumption that tree trunks can be expected to stand vertically, Huertas et al. [9] proposed a stereo-based system integrated with an edge detector method to estimate the diameters of trees and based on that to construct a tree traversability image for autonomous off-road navigation. Differently, Juman et al. [5] used this vertical assumption in combination with color space and the Microsoft Kinect sensor to detect tree trunks of oil-palm plantations.

In addition to ground-based robots, the ability to operate autonomously with minimal human assistance are also investigated in low-flying aircraft, such as in unmanned aerial vehicle (UAV). Roberts et al. [10] formulated a model-based tree detector, making use of motion saliency, to limit mapping to trees that are nearby to their aerial vehicle. Ortega et al. [11] presented a methodology for autonomous navigation of UAV with detection and evasion of trees by using a deep neural network approach, and Jiang et al. [12] presented an autonomous flight method that identifies obstacles and performs trajectory mapping to be used in precision forestry applications.

Typically, vision-based systems that use stereo vision approaches need to deal with feature-matching processes which can fail on unstructured terrain [13] and can be inaccurate due to possible parallax errors [7]. Thus, an appropriate approach is required to deal with challenges in stereo correspondences which can be complex due to some digital image issues as photometric distortions and noise, foreshortening, perspective distortions, ambiguous patterns, and occlusions and discontinuities [14].

The correspondence problem is at the core of each stereo algorithm, and because of that it has been extensively researched [15]. Yoon and Kweon [16] introduced an adaptive support weight stereo matching based on photometric and geometric relationship. Gerrits and Bekaert [17] presented a stereo matching algorithm which uses a color segmentation process to reduce the influence of outliers in the window-based aggregation process. Laureano and Paiva [18] considered a multi-resolution analysis that uses images pyramids and perceptual grouping weight to present a stereo method based on adaptive-windows. Mattoccia [15] underlined the geometric and photometric structure of a scene to formulate multiple assumptions for local consistency. Furthermore, stereo vision methods were evaluated to present a comparison among them [19], [20], [21].

Determining correspondences in two or more images is a challenging task in both indoor and outdoor environments. However, the uncontrolled nature of external ambiances increases this task and stereo vision methods may not work properly. The assumption that corresponding pixels are in the stereo images pair can be affected by soft or drastic changing in lighting, shadows, sudden camera movement due to irregular terrain, and by repetitive patterns, which are very common in forest environments. Taking these issues into account, we propose a method to be applied in the context of external environments. Two main requirements were formulated: (1) the method must detect tree trunk and (2) it must preserve the tree design (shape and boundaries).

III. Proposed Method

In our study, we considered complex scenes where tree trunks are prominent obstacles. The proposal begins with the

detection of trees to thus delineate their design. The input image, I, is divided into small pieces and each of them goes through contrast models that distinguish objects based on their color and brightness. 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 [1] to create models that extract vertical features of trees. Eq. 1 reproduces this contrast model function.

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

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, $F = (B * I)$, to produce curves of contrasting areas. Each model filter produces its own curve through the summed-up value of each column of F, Eq. 2.

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

$curve$ 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. 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. Besides, a threshold, $0 < \alpha \leq 1$, is applied to accept only valleys at a specific depth. If the α is set to a higher value, then it accepts more curves. In the tests, α was set the value 0.4.

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) [22]. In the tests, we used models with size 8, 14, 22, 44, 64, and 84.

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).

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

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

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

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, σ. 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.

Furthermore, 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 $\overline{U}$.

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

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 [23]).

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

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 (th) that discards non-tree pixels. Eq. 8 checks if points of I_s are in accordance with this threshold (receive the value 1).

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

where $th_1 = \lambda_1 m'$ and $th_2 = \lambda_2 m''$, if the values of λ_1 and λ_2 increase, then it will accept more bright and green points. In the experimental tests, we set them with values 5 and 1.3, respectively. Points of I_s at index i that receive the value 0 are then discarded.

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.

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

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.

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

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

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

Furthermore, as 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 γ. In the tests, γ was set with the values 0.2 or 0.4.

IV. Experimental Results

The evaluation of the proposed method was divided into two phases. The first one evaluates the quality of our method in keeping parts of trees and removing non-tree parts. The second one evaluates the use of this method by stereo vision approaches in calculating the disparity of trees. To perform these evaluations, we prepared an original dataset with images collected from rural and semi-rural environments. We used a monocular RGB camera, and for each scene, we took two shots shifted by a horizontal movement of the camera, which provided a stereo image dataset[1] with 11 corresponding image pairs.

These images were downsampled to 384×288 and carefully segmented by hand to allow the analysis of the results, acting as reference images in the experiments. In addition, for each image pair, two ground truth disparity maps were prepared by following these steps: (1) we calculate the center of mass from both segmented images; (2) we measure the distance between these two points in the $x-coordinate$; (3) the distance result was used to set the disparity value for all points in the tree segment; and (4) a tree disparity map was yielded considering the tree position in both images; thus two disparity maps were prepared. Fig. 1 shows a sample of this image dataset.

A. Quality evaluation

In the methodology explained in Section III, the main objective is to obtain the segments of trees that are in a scene. An image quality evaluation allows to measure how the method reach this task. Four different metrics were considered to evaluate the degradation of the trees by the proposed method in comparison to the images that were manually segmented. We used the Structural Similarity Index (SSIM), Complex-Wavelet Structural Similarity (CW-SSIM), Normalized Cross-Correlation (NCC), and Peak Signal to Noise Ratio (PSNR).

[1]The dataset is available on https://github.com/vicom-ifgo-urutai/datasets/blob/master/dataset_iscc2019.zip

(a) (b) (c) (d)

Fig. 1. A sample of the image dataset. Stereo image pairs, $I^{(1)}$ and $I^{(2)}$, are shown in column (a). The stereo segmented images are shown in column (b). Column (c) shows the centers of mass that were used to define the disparity maps (d).

With the exception of the last one, the output of the first three metrics varies in a standard range from 0 to 1. PSNR metric was set to vary from 0 to 48.13. Outputs close to the maximum value mean a better result.

To run the experiment, we divided the stereo image pairs into two groups: $I_s^{(1)}$ which contains the segmented images of the left camera; and $I_s^{(2)}$ which contains the segmented images of the right camera. In TABLE I, the presented values represent the average of each group. The results indicate that the proposed method in this article reached important outcomes as the SSIM greater than 0.9.

TABLE I
Quality Measures.

	SSIM	CWSSIM	NCC	PSNR
$I_s^{(1)}$	0.9188	0.8584	0.8240	27.58
$I_s^{(2)}$	0.9167	0.7932	0.8720	26.75

Furthermore, we evaluate the hits and fails of the proposed method by calculating the true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), and the accuracy (ACC). TABLE II shows these statistical results in which TPR and TNR obtained more than 77% assertiveness, and few errors, according to FPR and FNR. Besides, the accuracy was greater than 96% in both image groups.

TABLE II
Statistical Measures.

	TPR	TNR	FPR	FNR	ACC
$I_s^{(1)}$	0.7795	0.9896	0.2205	0.0104	0.9683
$I_s^{(2)}$	0.7781	0.9841	0.2219	0.0159	0.9627

Fig. 2 shows the visual results of the proposed method. In the first column, a pair of input images is presented, and in the second column, their respective ground truth images. 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.

(a) (b) (c) (d)

Fig. 2. Results from the proposed method. A stereo image pair is 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).

B. Disparity evaluation

Firstly, we used the Bad-Pixel Error as a quantitative measure to estimate the errors from the computed disparity maps. It computes the error between a computed disparity map $D(x, y)$ and its respective ground truth map $T(x, y)$, as given by Eq. 12.

$$B = \frac{100}{N} \sum_{(x,y)} (|D(x, y) - T(x, y)| > \delta) \tag{12}$$

where N is the total number of pixels and δ is a disparity error tolerance (in this work $\delta = 1$).

The segmented images, $I_s^{(1)}$ and $I_s^{(2)}$, were used as inputs in eight different stereo vision methods and their outputs were compared with the ground truth maps employing the considered measure. We calculated the disparity maps in two matching ways, i.e., one that considers the image $I_s^{(1)}$ as the reference image ($I_s^{(1)} \rightarrow I_s^{(2)}$) and another that considers $I_s^{(1)}$ as the target image ($I_s^{(1)} \leftarrow I_s^{(2)}$). Thus, we yielded two groups of disparity maps, $D_1^{(1)}$ and $D_2^{(2)}$, to be compared with their corresponding groups of ground truth maps, $T_1^{(1)}$ and $T_2^{(2)}$.

In this evaluation, we used the following stereo vision methods: Fixed Window (FW), Shiftable Window (SW) [24], Large Occlusion (LO) [25], Bilateral support weights (BL and BLnoSpatial) [21], Multi-resolution and Perceptual Grouping (MRPG) [18], Guided filter support weights (GF) [26], and Segmentation-based stereo method (SB) [17].

TABLE III summarizes the results. The MRPG method reached the lowest error in the first column while the best performance in column two was reached by the BLnoSpatial method. Despite this note, it is clear that the disparity calculation error is almost the same for all methods, which means that even simple methods such as FW and SW can do very well compared to more robust methods such as BL. Fig. 3 shows some samples of the disparity maps yielded by the considered stereo methods.

We also calculated the overlap score between the yielded disparity maps and their ground truth maps, $o(D, T) = n(D \cap T)^2/(n(D)n(T))$, [1]. Three parameters were defined: $Tree$, $Non - Tree$, and ALL. The first one considers only the tree area, while the second one considers only the non-tree area. Finally, the last one considers all points of the image

TABLE III
DISPARITY MAP EVALUATION.

	$I_s^{(1)} \rightarrow I_s^{(2)}(\%)$	$I_s^{(1)} \leftarrow I_s^{(2)}(\%)$
FW	8.26	8.84
SW	7.90	8.24
LO	9.55	10.0
BL	7.31	7.55
BLnoSpatial	7.25	7.53
MRPG	6.98	7.72
GF	7.42	7.71
SB	7.61	8.23

(a) (b) (c) (d)

Fig. 3. Disparity map results. The original images in column (a). The segmented images in column (b). The yielded disparity maps in column (c), and the ground truth maps in (d). From rows 1 to 4, it presents the results from the methods LO, BL, MRPG, and SB, respectively.

to be compared with the hand-labeled image of the dataset. Considering the average of the two groups, the $overlap$ was greater than 0.9 in the parameter ALL, which indicates that tree and non-tree points were well-segmented from each other. On the other hand, in the $Tree$ parameter, the results were less than 0.7, which shows that portions of tree were missed (TABLE IV).

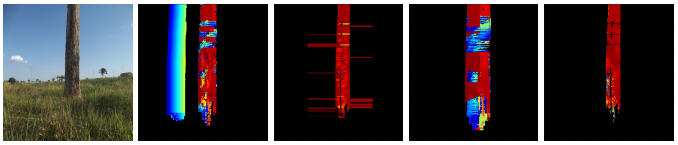
TABLE IV
OVERLAP EVALUATION.

	$Tree$	$Non - Tree$	ALL
$o(D_1^{(1)}, T_1^{(1)})$	0.69	0.96	0.93
$o(D_2^{(2)}, T_2^{(2)})$	0.66	0.95	0.92

Despite of this, we also applied the same strategy that were used to obtain the ground truth maps. Thus, the center of mass of each segment was used to estimate the projection of the tree in the other image plane. A Bad-Pixel error of 8.25% was obtained which is close to the results presented in TABLE III, but in this case, with a simpler approach.

Furthermore, the proposed method can deal with some out-

comes produced by stereo vision methods, as *ghost artifacts*[2], *streaking effect*[3], and *foreground fattening*[4]. In Fig. 4 these artifacts are presented and they can be easily removed since the area of the tree is known. In this way, points on the disparity map which are outside of the tree segment can be discarded by conserving only the points in the segment.



(a) (b) (c) (d) (e)

Fig. 4. Disparity map artifacts. In column (a), the input image. Columns (b), (c), and (d) show samples of the ghost artifact from SW, streaking effect from LO, and foreground fattening from FW. The final result in column (e) was obtained from BL method.

V. Conclusion and Future Work

In this paper, we introduce a new method for segmenting tree trunks from their backgrounds based on image color information and contrast models. We highlight that outdoor environments are complex to model due to natural constraints that are difficult to predict, as adverse weather conditions and frequent changes in lighting. Thus, they require efficient approaches for the development of computational solutions to be used in practical applications, such as in smart farming demands.

In our experiments, the results showed that the proposed method preserves contours and shape of the trees to ensure their design, finding the areas of trees and detaching them from non-tree areas. In this way, it can reach suitable results in disparity calculation.

As the next phase of the work, we intend to increase the dataset with the inclusion of scenes from dense forests, as well as the incorporation of image salience techniques to detect the most prominent tree in such scenes.

References

[1] 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.

[2] X. Chen, S. Wang, B. Zhang, and L. Luo, "Multi-feature fusion tree trunk detection and orchard mobile robot localization using camera/ultrasonic sensors," *Computers and Electronics in Agriculture*, vol. 147, pp. 91–108, 2018.

[3] 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.

[4] C. Cabo, C. Ordóñez, C. A. López-Sánchez, and J. Armesto, "Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 69, pp. 164–174, 2018.

[2]Points not presented in the original stereo pair that appear mainly due to large parallax between the adjacent images.

[3]It results from the lack of coherence in the vertical direction when a 1-D optimization problem is applied.

[4]Incorrect estimations near the objects boundaries caused by pixels at different depths.

[5] 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.

[6] G. Reina, A. Milella, and R. Rouveure, "Traversability analysis for off-road vehicles using stereo and radar data," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, March 2015, pp. 540–546.

[7] W. Ali, F. Georgsson, and T. Hellstrom, "Visual tree detection for autonomous navigation in forest environment," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 560–565.

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

[9] A. Huertas, L. Matthies, and A. Rankin, "Stereo-based tree traversability analysis for autonomous off-road navigation," in *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, vol. 1. IEEE, 2005, pp. 210–217.

[10] R. Roberts, D.-N. Ta, J. Straub, K. Ok, and F. Dellaert, "Saliency detection and model-based tracking: a two part vision system for small robot navigation in forested environment," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8387, pp. 27–, 05 2012.

[11] S. Dionisio-Ortega, L. O. Rojas-Perez, J. Martinez-Carranza, and I. Cruz-Vega, "A deep learning approach towards autonomous flight in forest environments," in *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Feb 2018, pp. 139–144.

[12] S. Jiang, K. A. Stol, W. Xu, and B. Graham, "Towards autonomous flight of an unmanned aerial system in plantation forests," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 911–919.

[13] H. Balta, G. De Cubber, D. Doroftei, Y. Baudoin, and H. Sahli, "Terrain traversability analysis for off-road robots using time-of-flight 3d sensing," in *7th IARP International Workshop on Robotics for Risky Environment-Extreme Robotics, Saint-Petersburg, Russia*, 2013.

[14] S. Mattoccia, "Stereo vision: Algorithms and applications," 2013. [Online]. Available: http://www.vision.deis.unibo.it/smatt/stereo.htm

[15] ——, "A locally global approach to stereo correspondence," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1763–1770.

[16] K.-J. Yoon and I.-S. Kweon, "Locally adaptive support-weight approach for visual correspondence search," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 924–931 vol. 2.

[17] M. Gerrits and P. Bekaert, "Local stereo matching with segmentation-based outlier rejection," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*. IEEE, 2006, pp. 66–66.

[18] G. T. Laureano and M. S. V. de Paiva, "Disparities maps generation employing multi-resolution analysis and perceptual grouping," in *2008 First Workshops on Image Processing Theory, Tools and Applications*, Nov 2008, pp. 1–6.

[19] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[20] G. Vieira, F. Soares, N. Sousa, J. Gil, R. Parreira, G. Laureano, R. Costa, and J. Ferreira, "Stereo vision methods: from development to the evaluation of disparity maps," in *Proceedings of XIII Workshop de Visão Computacional*, 2017.

[21] A. Hosni, M. Bleyer, and M. Gelautz, "Secrets of adaptive support weight techniques for local stereo matching," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 620–632, 2013.

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

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

[24] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[25] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[26] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, Feb 2013.