Vine trunk detector for a reliable robot localization system

Jorge Mendes[a], Filipe Neves dos Santos[b], Nuno Ferraz[a], Pedro Couto[c], Raul Morais[a]
[a]INESC TEC - INESC Technology and Science and Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
[b]INESC TEC - INESC Technology and Science and Faculty of Engineering, University of Porto, Porto, Portugal
[c]CITAB - Center for the Research and Technology of Agro-Environmental and Biological Sciences and
Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
e-mail: jorge_mendes_1@hotmail.com, fbsantos@inesctec.pt, andreferraz16@gmail.com, pcouto@utad.pt, rmorais@utad.pt

***Abstract*—Develop ground robots for crop monitoring and harvesting in steep slope vineyards is a complex challenge due to two main reasons: harsh condition of the terrain and unstable localization accuracy got from Global Positioning Systems (GPS). For this context, a reliable localization system requires a high density of natural/artificial features and an accurate detector. This paper presents a novel visual detector for Vineyards Trunks and Masts (ViTruDe). The ViTruDe detector was developed considering the constrains of a cost-effective robot to carry-out crop monitoring tasks in steep slope vineyard environment. The obtained results with real data shows an accuracy higher than 95% for all tested configurations. The training and test data are made public for future research work. This approach is a contribution for an accurate and reliable localization system that is GPS-free.**

I. Introduction

Develop ground robots for crop monitoring and harvesting is a complex challenge because robotic sensing, perception and interpretation of the crop needs to be efficient, accurate, and robust in an unstructured environment [16].

The strategic European research agenda for robotics [12] states that robots can improve agriculture efficiency and competitiveness. However, few commercial robots for agricultural applications are available [16].

Fig. 1. A typical steep slope Terraced Vineyard in the Douro region of Portugal.

In Europe space few European funded projects are developing monitoring robots for flat vineyards: the VineRobot [17] and Vinbot [18]. However, vineyards built on steep slope hills presents an higher complex environment for the machinery and automation development. These called steep slope vineyards exist in Portugal in the Douro region - an UNESCO heritage place - Fig. 1, and in other regions of five European countries. The context of a vineyard built in a steep hill presents several robotics challenges:

- Terrain characteristics produces signal blockage and multi-reflection which decreases the availability and accuracy of the Global Positioning System (GPS). Therefore, GPS is not always available and reliable;
- Harsh conditions of the terrain limits the accuracy of dead-reckoning sensors - for example odometry and inertial measurement systems (IMU);
- Terrain slopes imposes constrains to the robot path planning. Safe robot motion planning and control depends on accurate maps of the vineyard and precise information about its posture (localization and attitude).

In these vineyards, the high elevation mask imposed by the hills limits the number of GPS satellites in view [13]. Thus, GPS accuracy is low and robot localization is not always available. Beside, a GPS receiver can suffer spoofing attacks which is a possible safety issue. For these reasons, a redundant localization solution is needed to get a full-time available, efficient and safe robot.

VineSLAM [13] [14] is GNSS-free (Global Navigation Satellite System) localization system for steep slope vineyards. VineSLAM considers a hybrid map architecture to increase the localization accuracy and robustness. The topological localization of the robot is updated considering artificial landmarks, such as those proposed in [15]. In metric localization, VineSLAM considers natural vineyard features [13] to increase the system redundancy and reliability. These natural features are the vine trunks and masts. They exist in all vineyards with a large density. However, a reliable detector of these natural features requires several sensors information, such laser range finders (LRF), cameras and/or RGB-D sensors. The use of standalone LRF-based detectors is not reliable because grass and dust creates small artifacts that are classified as trunks/masts [14]. Vision-based detectors provides com-

978-1-5090-2255-7/16 $31.00 © 2016 IEEE
DOI 10.1109/ICARSC.2016.68

plementary and richer information to reach a reliable natural feature detector. This work presents a vision-based detector for these natural features detection.

In this paper, section II presents the related work to the outdoor simultaneous localization and mapping (SLAM) problem. Section III presents ViTruDe approach for vineyards natural features detection. Section IV presents the results obtained with ViTruDe under real and simulated data. Section V presents the paper conclusions.

II. Related Work

In robotics, the problem of autonomous navigation is widely explored due to its importance in the design of intelligent mobile robots. Depending on the environment and robotic application a solution for simultaneous localization and mapping (SLAM) problem is almost always needed. The appearance and subsequent development of probabilistic techniques described in [3] marked a point of reference in the construction of methods for the SLAM problem [1] [2]. The SLAM problem for indoor scenarios is largely explored and several approaches in the literature can be found. In [4], Extended Kalman Filter (EKF) was considered at the time the most used method to solve the SLAM problem. The system state vector of EKF-based SLAM contains both: the pose of the vehicle - its position and orientation within the environment - and parameters that describes the features on that environment. Several variants, based on features-based maps are Unscented Kalman Filter (UKF) SLAM, EKF-based FastSLAM version 2.0, and UKF-based FastSLAM (uFastSLAM), are benchmarked in [5].

Unlike indoor scenarios, agriculture scenarios are less structured, more dynamic and sensors are exposed to several adverse conditions. This hampers detection and extraction of natural/artificial features, required for SLAM. [9] uses image key-points (low level features) to estimate a visual-based odometry, which allows the use of these key-points on 3D mapping. However, in agricultural scenarios, the majority of image key-points (low level features) are associated with leafs and grass. Wind moves leafs and grass highly reduces localization accuracy. So, it is needed to use high level features concept [11] to increase the system reliability. [10] selects rocks as high level features for localization of rovers for space exploration. In [6], an EIF-SLAM based approach is tested in olive groves. In this work, olive stems of the plantation are detected for mapping and localization. The olive stems are acquired by a range sensor laser and a monocular vision system. A support vector machine (SVM) implemented on the vision system detects olive stems in the images acquired from the environment. Two main issues are reported, outliers on feature detection and wrong closed-loop association. In [7], an agricultural robotics survey presents the state of the art to the questions *Which is the biological feature of interest?* and *How is such a feature extracted/detected?*. However, this survey focus orchards and vegetable farm environments. A vineyard has different natural features [8]. In steep slope vineyards the challenge is more complex and unexplored. Remember, GPS is not always accurate and available to help at least at the mapping stage. So, a reliable detection system for natural vineyard features is needed for an accurate, robust and redundant robot localization system.

III. ViTruDe approach

ViTruDe is the proposed approach to detect vineyards mast and trunks. In VineSLAM, this detector is fundamental to detect the features used by features based map.

ViTruDe has three main components: Keypoints Extractor; Keypoints Search and Region Descriptor Extractor; and SVM classification (Fig. 2).

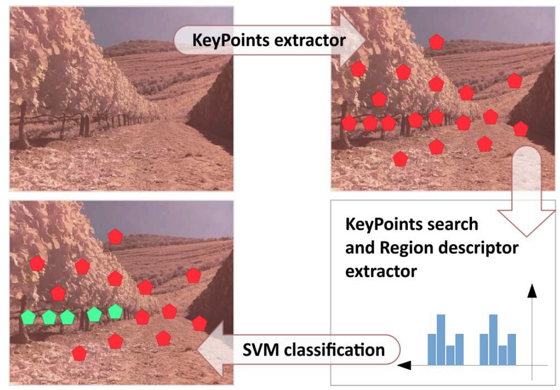

Fig. 2. Information flow and main components of ViTruDe.

ViTruDe first detects key-points in the input image, where a small region will be analyzed for the presence of trunks. This component reduces the computational cost to detect natural vineyard features in all images. In this work, for key-points extraction, the Sobel operator was the adopted approach.

ViTruDe extracts a descriptor in the region around each key-point. This descriptor is the input of a support vector machine (SVM) classifier. Based in the training step, the SVM is able to classify if a mast or trunk is present in that small image window. The region has a fixed size of 100×30 pixels. This size was selected to capture a standard trunk at a distance of 1-2 meters from a camera with 640×480 pixels and fill 60-90% of the image region.

The proposed descriptor is based in the local binary pattern (LBP) codes. LBP is a powerful gray-level invariant texture primitive.

The non-parametric LBP operator was firstly mentioned by Harwood et al [19], then introduced by Ojala et al.[20] for textured image description.

The original LBP works in a grid size of 3×3 pixels for a given arbitrary pixel over an input gray-scale image. The LBP code is computed by comparing the gray-level value of the center pixel and its neighbors, within the respective grid. It should be pointed out that gray-value of the neighbor pixels, which are not covered by the grids, is estimated by interpolation. The thresholding stage is carried out with respect to the center pixel, resulting in a binary number called LBP code. To describe the image texture, a LBP histogram (hLBP) is built from all binary patterns of each image pixel:

$$H(k) = \sum_{m=1}^{M}\sum_{n=1}^{N} f(\mathrm{LBP}_{P,R}(m,n),k), \quad k \in [0,K] \tag{1}$$

$$f(x,y) = \begin{cases} 1, x = y \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

where K is the maximal LBP pattern value.

Based on hLBP, two types of descriptors are constructed: *hLBP by colour* and *hLBP plus colour*, Fig. 3.

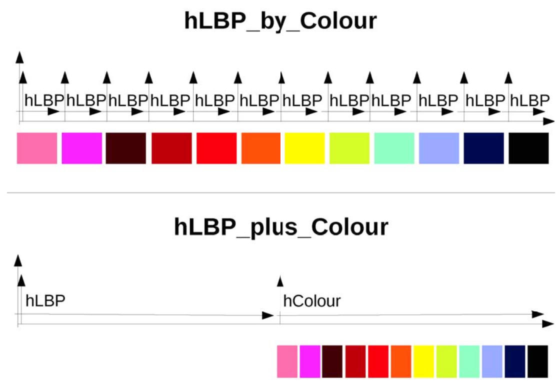

Fig. 3. The two descriptors selected for vineyards features description.

The *hLBP by colour* has one LBP histogram per colour space. The colour space is discretized into n colours. The length of *hLBP by colour* is given by the number of LBP codes times the number of colour spaces. To reduce descriptor size, the LBP uniform variant is selected with 8 colour spaces. Considering $LBP^{u}_{8,2}$, the descriptor size has $59 \times 8 = 472$ bins. Considering this descriptor, the extractor in each pixel detects the related colour space and increments the histogram bin related to the LBP code extracted for that pixel.

In contrast, *hLBP plus colour* has two histograms, one for LBP codes and one for colour spaces. The colour space is discretized into n colours. The length of *hLBP plus colour* is given by the number of LBP codes plus the number of colour spaces. For this descriptor the $LBP_{8,2}$ is selected with 8 colour spaces. So, the descriptor size is $256 + 8 = 264$ bins. Considering this descriptor, the extractor in each pixel detects the related colour space and increments the histogram bin related to that colour. Then, the pixel LBP code is extracted to increment the histogram bin related to that code.

IV. Tests and Results

The ViTruDe approach, train, test and validation was realized with real data acquired by the AGROB V14 robot [13]. This data is made public at http://agrob.inesctec.pt. These data were acquired in three different days with clear sky, after sunrise and before the sunset.

For SVM training and testing, 6 different data-sets where built. They were built considering images frames from three different PiCamera filter configurations: NoIR without filter, NoIR with filter and a RGB camera. The camera resolution is 640×480 pixels with a framerate of 5 fps. For each camera filter configuration, nearly 200 image frames where collected from ROS bag acquired by AGROB V14. From these images, sub-images of 100×30 pixels where collected to obtain 3000 for each class.

To validate SVM performance under multiple classes, the six data-sets were divided in two groups.

- three data-sets were built considering two classes (Trunks and Others); and,
- other three were built considering four classes (Trunks, Sky, Floor and Others).

Table I shows a summary for each data-set built. Fig. 4 shows a snapshot of some images stored in that six data-sets.

TABLE I
Description of data-sets built.

Name	Classes	N images/class	train/test
NoIRF_2	Trunks and Others	3000	2500/500
NoIRI_2	Trunks and Others	3000	2500/500
RGB_2	Trunks and Others	3000	2500/500
NoIRF_4	Trunks, Sky, Floor and Others	3000	2500/500
NoIRI_4	Trunks, Sky, Floor and Others	3000	2500/500
RGB_4	Trunks, Sky, Floor and Others	3000	2500/500

Several tests were realized considering these six data-sets and the two types of ViTruDe descriptors. Tests results are summarized in Tables II and III. These tests were realized in a laptop with Intel Quad Core i7-4700HQ (@2.4 GHz) processor, 12 GB of RAM, and Ubuntu 14.04 LTS.

The ViTruDe SVM implementation is based in the open source LibSVM library. Three SVM kernels were tested: Linear (LINEAR), Radial Based Function (RBF) and Polynomial (POLY).

The RBF kernel has the parameter γ that can be configured for better results. Several values were tested, however the default value led to the best results. The POLY kernel has by default the same γ and a $Coef0 = 0$.

The default value of γ is $\gamma = \frac{1}{n_{\mathrm{features}}}$, which means:

- for *hLBP by colour* $\gamma = 0.00212$; and,
- for *hLBP plus colour* $\gamma = 0.00379$.

Table II presents the results obtained when two classes are considered. The column *Results Accuracy* has 4 sub columns:

- T, where are presented the number of images with trunks that were classified as trunks;
- TN, where are presented the number of images with trunks that were classified as other;
- O, where are presented the number of images without trunks that were classified as others;
- ON, where are presented the number of images without trunks that were classified as trunks.

The column *Frame Processing Time* presents the processing time of ViTruDe in each video frame. In this test, for key-points extraction, the Sobel operator was the adopted approach. First we extract the edges. After, to reduce even further the processing time, the key-points are selected in that edges

Fig. 4. Samples from the database acquired for SVM train and test. In rows: first line sky, second others, third floor, and forth mast and trunks. In columns: first set (15 images) NoIR Camera without filter, second set NoIR Camera with filter, third set RGB camera.

TABLE II
TESTS PERFORMED WITH TWO CLASSES (TRUNKS AND OTHERS).

	Descriptor (Type)	SVM (Kernel)	Results Accuracy				Accuracy (%)	Frame Processing Time (sec/frame)
			T	TN	O	ON		
NoIR Without Filter	hLBP_by_Colour	LINEAR	483	17	478	22	96.10	0.181962
		RBF	485	15	477	23	96.20	0.411859
		POLY_C_0	484	16	478	22	96.20	0.207692
		POLY_C_100	485	15	479	21	96.40	0.188917
	hLBP_plus_Colour	LINEAR	474	26	480	20	95.40	0.134353
		RBF	493	7	488	12	98.10	0.272993
		POLY_C_0	490	10	489	11	97.90	0.144154
		POLY_C_100	488	12	484	16	97.20	0.130935
NoIR With Filter	hLBP_by_Colour	LINEAR	487	13	488	12	97.50	0.235439
		RBF	497	3	490	10	98.70	0.589698
		POLY_C_0	492	8	493	7	98.50	0.366966
		POLY_C_100	490	10	494	6	98.40	0.259135
	hLBP_plus_Colour	LINEAR	487	13	491	9	97.80	0.138574
		RBF	490	10	497	3	98.70	0.320033
		POLY_C_0	493	7	494	6	98.70	0.149617
		POLY_C_100	492	8	494	6	98.60	0.129153
RGB	hLBP_by_Colour	LINEAR	496	4	494	6	99.00	0.208092
		RBF	499	1	498	2	99.70	0.401960
		POLY_C_0	498	2	498	2	99.60	0.240220
		POLY_C_100	497	3	497	3	99.40	0.195429
	hLBP_plus_Colour	LINEAR	495	5	497	3	99.20	0.142418
		RBF	496	4	500	0	99.60	0.327056
		POLY_C_0	498	2	499	1	99.70	0.144257
		POLY_C_100	496	4	498	2	99.40	0.151497

but only those spaced by 30 pixels. The processing time is obtained by the average time of processing 100 frames.

Table III presents the results obtained when four classes are considered. The column *Results Accuracy* has 8 sub columns:

- T, where are presented the number of images with trunks that were classified as trunks;
- TN, where are presented the number of images with trunks that were classified as other class;
- F, where are presented the number of images with floor that were classified as floor;
- FN, where are presented the number of images with floor that were classified as other class;
- S, where are presented the number of images with sky that were classified as sky;
- SN, where are presented the number of images with sky that were classified as other class;
- O, where are presented the number of images without trunks/sky/floor that were classified as others;
- ON, where are presented the number of images without trunks/sky/floor that were classified as other class.

The column *Frame Processing Time* presents the processing time of ViTruDe in each video frame. This test was performed under the same conditions as reported for Table II.

Comparing table II and III is possible to conclude that SVM has a better accuracy when multiple classes are considered. However, the frame processing time is almost two times higher when we compare the multiple classes approach with the single class approach.

The Fig. 5 presents the confusion matrix of the best and worst configuration for tests performed with four classes. Worst result corresponds to the test with NoIR Without Filter camera, *hLBP by Colour* descriptor and RBF kernel. Best

TABLE III
TESTS PERFORMED WITH FOUR CLASSES (TRUNKS, FLOOR, SKY AND OTHERS).

	Descriptor (Type)	SVM (Kernel)	Results Accuracy								Accuracy (%)	Frame Processing Time (sec/frame)
			T	TN	F	FN	S	SN	O	ON		
NoIR Without Filter	hLBP_by_Colour	LINEAR	489	11	499	1	500	0	485	15	98.65	0.412045
		RBF	487	13	497	3	500	0	480	20	98.20	0.951266
		POLY_C_0	483	17	499	1	500	0	484	16	98.30	0.656236
		POLY_C_100	488	12	499	1	500	0	486	14	98.65	0.494610
	hLBP_plus_Colour	LINEAR	480	20	499	1	500	0	493	7	98.60	0.290738
		RBF	493	7	500	0	500	0	494	6	99.35	0.617613
		POLY_C_0	488	12	500	0	500	0	495	5	99.15	0.308523
		POLY_C_100	489	11	499	1	500	0	496	4	99.20	0.294638
NoIR With Filter	hLBP_by_Colour	LINEAR	492	8	500	0	500	0	495	5	99.35	0.358088
		RBF	494	6	500	0	500	0	491	9	99.25	0.757214
		POLY_C_0	495	5	500	0	500	0	495	5	99.50	0.471825
		POLY_C_100	493	7	500	0	500	0	495	5	99.40	0.354391
	hLBP_plus_Colour	LINEAR	494	6	500	0	500	0	496	4	99.50	0.246987
		RBF	492	8	500	0	500	0	499	1	99.55	0.582459
		POLY_C_0	495	5	500	0	500	0	498	2	99.65	0.238455
		POLY_C_100	494	6	500	0	500	0	497	3	99.55	0.231434
RGB	hLBP_by_Colour	LINEAR	497	3	497	3	500	0	500	0	99.70	0.297905
		RBF	498	2	497	3	500	0	498	2	99.65	0.685880
		POLY_C_0	498	2	497	3	500	0	500	0	99.75	0.384702
		POLY_C_100	499	1	498	2	500	0	500	0	99.85	0.306794
	hLBP_plus_Colour	LINEAR	500	0	499	1	500	0	499	1	99.90	0.262131
		RBF	499	1	498	2	500	0	498	2	99.75	0.580006
		POLY_C_0	500	0	499	1	500	0	500	0	99.95	0.277735
		POLY_C_100	500	0	499	1	500	0	499	1	99.90	0.291064

result corresponds to the test with RGB camera, *hLBP plus Colour* descriptor and Poly kernel with a $Coef0 = 0$.

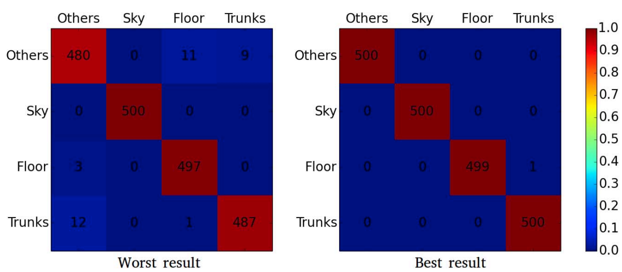

Fig. 5. Confusion matrix of the best and worst configuration for tests performed with four classes.

When we compare the accuracy of the two types of ViTruDe descriptors, in average both approaches have the same accuracy under the same camera configuration and number of SVM classes. However, the frame processing time is 25% faster when the *hLBP plus colour* is used. This happens because the *hLBP plus colour* descriptor has almost half the size of the *hLBP by colour* descriptor.

When we compare the accuracy of ViTruDe under different camera filter configuration, it is noticed that RGB PiCamera has the best accuracy in average. However, a PiCamera without IR filter and with a special filter for Normalized Difference Vegetation Index (NDVI) image extraction has similar accuracy. This is important because a single camera (for NVDI) can be used for both application uses and localization perception.

The best accuracy (99.90%) was reached by the hLBP plus colour descriptor, under multiple SVM classes with linear kernel and a RGB camera. With this configuration the frame processing time was 262 ms.

In contrast, the best frame processing time (129 ms) was reached by the hLBP plus colour descriptor, under single SVM class with Poly kernel and a RGB camera. With this configuration the SVM accuracy was 98.60%.

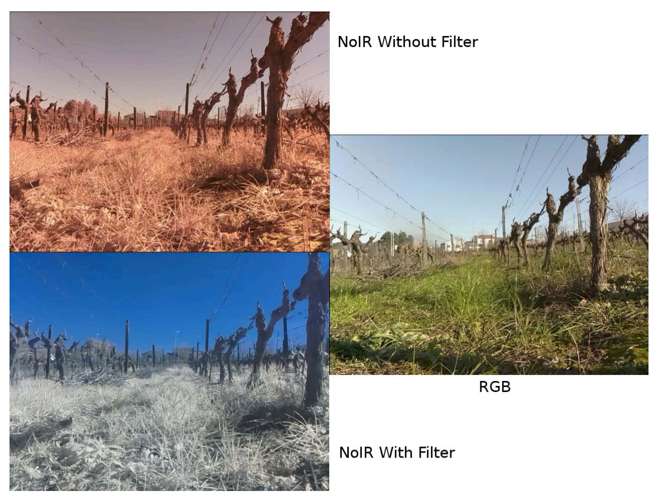

Fig. 6. Input samples images.

For the video processing tests, all camera filter and descriptor configurations were used. Fig. 6 shows the three types of input images. Fig. 7 shows the result output obtained by ViTruDe with hLBP plus colour descriptor and multiple SVM classes with linear kernel.

The results obtained by ViTruDe, under data acquired by AGROB V14, can be found at:

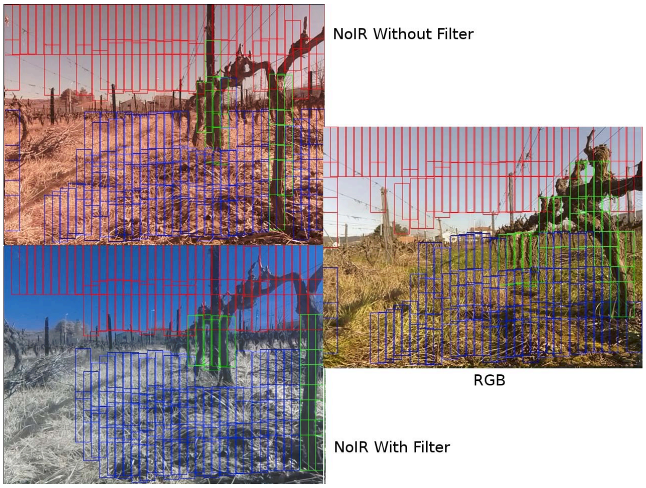

Fig. 7. Output results obtained by *hLBP plus colour* with four classes. Red rectangles classified as sky, green as mast/trunks, blue as ground.

- 2 Classes - SVM Kernel Linear - Descriptor *hLBP by colour* - https://youtu.be/Mg-ywzm7mSM
- 2 Classes - SVM Kernel Linear - Descriptor *hLBP plus colour* - https://youtu.be/4NZ5G7pBsZY
- 4 Classes - SVM Kernel Linear - Descriptor *hLBP by colour* - https://youtu.be/uqSfXBn8JuY
- 4 Classes - SVM Kernel Linear - Descriptor *hLBP plus colour* - https://youtu.be/kfp0b8uKtsM

V. Conclusion

The proposed ViTruDe detector was developed considering the constraints of a cost-effective robot to carry-out crop monitoring tasks in steep slope vineyards. The obtained results with real data shows an accuracy higher than 95% for all ViTruDe tested configurations.

From field experiments, two main conclusions emerges. The average accuracy of the two proposed ViTruDe descriptors is similar but frame processing time is 25% faster when the *hLBP plus Colour* is used. This happens because the *hLBP plus Colour* descriptor has almost half the size of the *hLBP by Colour* descriptor. The accuracy of ViTruDe under different camera filter configuration has, in average, a variation of 5%. The ViTruDe with RGB PiCamera has the best accuracy. However, a PiCamera with a special filter for NDVI image extraction has similar accuracy. This is an important fact because a single camera (for NVDI) can be used for both application uses and robot localization.

As future work, ViTruDe will be integrated with VineSLAM approach allowing data fusion provided by ViTruDe and the LRF based detector [13]. Besides, a better key-point extractor will be researched. To prove the reliability of the proposed approach the database should be extended with data acquired on cloudy days and under artificial illumination. Also, we will deploy the ViTruDe source code to the raspberryPi GPU processing unit.

Acknowledgment

Project "NORTE-07-0124-FEDER-000060" is financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

[1] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. "FastSLAM: A factored solution to the simultaneous localization and mapping problem." AAAI/IAAI, 593-598, 2002.
[2] S. Thrun. "Robotic mapping: A survey." Exploring artificial intelligence in the new millennium, 1-35, 2002.
[3] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.
[4] Bailey, Tim, and Hugh Durrant-Whyte. "Simultaneous localization and mapping (SLAM): Part II." IEEE Robotics & Automation Magazine 13.3 (2006): 108-117.
[5] Kurt-Yavuz, Zeyneb, and Srma Yavuz. "A comparison of EKF, UKF, FastSLAM2. 0, and UKF-based FastSLAM algorithms." Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on. IEEE, 2012.
[6] Cheein, F. Auat, et al. "Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection." Computers and electronics in agriculture 78.2 (2011): 195-207.
[7] Auat Cheein, F., and Ricardo Carelli. "Agricultural robotics: Unmanned robotic service units in agricultural tasks." Industrial Electronics Magazine, IEEE 7.3 (2013): 48-58.
[8] Zaidner, Guy. "Multisensory Data Fusion for Autonomous Vineyard Sprayer Robot Navigation". PhD Thesis Ben-Gurion University of the Negev, 2014.
[9] Faessler, Matthias, et al. "Autonomous, Visionbased Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle." Journal of Field Robotics (2015).
[10] E. Boukas and A. Gasteratos and G. Visentin "Towards orbital based global rover localization." IEEE International Conference on Robotics and Automation (ICRA): 2874-2881, May 2015.
[11] Kostavelis, Ioannis and Gasteratos, Antonios "Semantic Mapping for Mobile Robotics Tasks." Robotics and Autonomous Systems Vol. 66C: 86-103, April 2015.
[12] euRobotics (2013). Strategic research agenda for robotics in Europe. Draft 0v42 [Online]. Available: http://ec.europa.eu/research/industrial_technologies/pdf/robotics-ppp-roadmap_en.pdf
[13] Neves Dos Santos, Filipe, et al, "Towards a Reliable Monitoring Robot for Mountain Vineyards" Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference,2015,April:37-43.
[14] Neves Dos Santos, Filipe, et al, "Towards a Reliable Robot for Steep Slope Vineyards Monitoring" Journal of Intelligent and Robotic Systems, Article 340, DOI: 10.1007/s10846-016-0340-5.
[15] Duarte, Marcos and dos Santos, Filipe Neves and Sousa, Armando and Morais, Raul, "Agricultural Wireless Sensor Mapping for Robot Localization", Robot 2015: Second Iberian Robotics Conference, 359-370, 2016, Springer International Publishing.
[16] Bac, C. Wouter, et al. "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead.", Journal of Field Robotics 31.6 (2014): 888-911.
[17] VineRobot - FP7 project. [Online]. Available: http://www.vinerobot.eu/
[18] Vinbot - FP7 project. [Online]. Available: http://vinbot.eu/
[19] Harwood, David and Ojala, Timo and Pietikäinen, Matti and Kelman, Shalom and Davis, Larry, "Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions", Pattern Recognition Letters,16,1,1–10,1995,Elsevier.
[20] Ojala, Timo and Pietikäinen, Matti and Harwood, David, "A comparative study of texture measures with classification based on featured distributions", Pattern recognition,29,1, 51–59,1996,Elsevier.