UNIVERSITÀ DEGLI STUDI DI TORINO

Scuola di Dottorato in Scienze della Natura e Tecnologie Innovative

Doctorate in Agricultural, Forest and Food Sciences

Cycle: XXXIII

New automation technologies and solutions for precision agriculture

Engr. Shahzad Zaman

Supervisors
Prof. Paolo Gay
Prof. Paolo Balsari

Ph.D coordinator
Prof. Eleonora Bonifacio

Years
2017 : 2020

Acknowledgements

I would like to express my gratitude to my supervisors Prof. Paolo Gay and Prof. Paolo Balsari, for the PhD research opportunity at DiSAFA, University of Torino, their support and guidance throughout my Ph.D. and always making themselves available in their busy schedules.

Further I would like to thank Dr. Lorenzo Comba for being a remarkable, amazing and a patient mentor, always helping and guiding me throughout my studies and teaching me, and Prof. Davide Ricauda Aimonino and Dr. Alessandro Biglia for their knowledge, help and guidance.

My thanks to entire DiSAFA team, for providing me with a friendly atmosphere and for their continuous help and guidance.

Moreover, I would like to thank;

My father Sher Zaman, for his hard work to provide me with a better life and education, and my mother Asiya Zaman for always taking care of me and praying for my success. My beloved wife Federica Rinaudo, for her honest friendship and a wonderful life partner, always being there to support me and cheer me up. My brothers Shafi-Ullah-Zaman and Shahid Zaman, for always being there to support and guide me, never letting me worry about anything. My sister Fozia Zaman, wishing she were here. My sisters Shazia Zaman and Sidra Zaman, for all the well wishes and support. All my friends with whom I spent wonderful and joyful time and their support and delightful memories.

A huge thank you to the University of Torino entire body for the opportunity to study in such prestigious university

Once again, I would like to thank all the amazing people from the Deepest Bottom of My Heart And for all the precious knowledge,
As Sir Francis Bacon quoted

‘Knowledge is Power’

Thesis abstract/summary

Agriculture plays a key role in sustaining and driving the world economy. It is important not only as provision of raw material for major industries but most importantly it is the source of global food supply. Over the next 40 years the global food system will be facing formidable challenges as the world population is increasing exponentially. Coping with future challenges will require more radical changes to the agricultural system and the development of new research providing new solutions to novel problems. Many current farming approaches will continue to compromise the global future capacity to produce, while contributing to the degradation of the environment and to climate change as well as the destruction of biodiversity, eventually leading to non-sustainability.

It is crucial that the future vision of agriculture identifies with sustainable intensification through systematic approach and integrating farming management concepts based on technological advances founded on engineering science. In fact, the advances made in agricultural engineering have delivered some of the most significant developments seen in modern farming.

In this context, precision agriculture (PA) is recognised as an essential approach to optimise crop-managing practices and to improve field products quality while ensuring environmental safety. The adaption of cutting edge, cost-effective technologies and new innovative solutions, aimed at making operations and processes more reliable, robust and economically viable, continues to be required. Robotics and automation technologies are playing a crucial role, with particular reference to unmanned vehicles for crop monitoring and site-specific operations. Autonomous ground and aerial vehicles can lead to favourable improvements in field operations, extending crop scouting to large fields and performing field tasks in a timely and effective way.

However, in complex agricultural scenarios, such as unstructured and irregular working environments, new innovative solutions linked to autonomous machines path planning and crop field mapping are required. Autonomous vehicles, in addition to proper knowledge of their instantaneous position, require an accurate spatial description of their environment, to perform infield tasks. The buildout of new reliable tools for mapping crop field are necessary for site specific management practices/planning.

In light of the above discussion, this thesis focuses on the development and implementation of innovative and systematic approaches to deal with complex agricultural challenges related to autonomous machines path planning and detailed crop field mapping (2D and 3D maps). In particular, the focus is on the 3D point cloud data analysis provided by sophisticated 3D remote sensing technologies, such as from imagery acquired by unmanned aerial vehicle (UAV) (processed using structure from motion approach) light detection and ranging systems (LiDAR), and by 3D depth cameras used to help control agricultural machines, hence allowing possible operations such as effective weed management with minimal pesticide, leading to providing advances in productivity, profitability and environmental sustainability.

During the research work a modelling framework has been developed to semantically interpret 3D point clouds of vineyards and to generate low complexity 3D mesh models of vine row. By reducing the number of instances required to describe the spatial layout and shape of vine canopies allows the amount of data to be drastically reduced while avoiding the loss of relevant crop shape information. The proposed methodology is able to process complex vineyard scenarios, such as curvilinear vine rows or missing plants autonomously.

The first step of this study is to explain the algorithm developed for the clustering and localisation of vine rows from raw 3D point cloud data. The following step explains the second algorithm which leads to an innovative modelling framework to generates low complexity 3D mesh models of vine rows. The valuable information provided by these 3D point clouds processing algorithms can be used for real time autonomous machines path planning and helping them execute their infield operations robustly and efficiently.

Finally, the development and implementation of an innovative and cost-effective monocular visual odometry system, properly calibrated for the localisation and navigation of tracked vehicles on agricultural terrains is presented. The proposed system helps to tackle the problems faced by GPS systems due to limitations and drawbacks when the satellite signal is poor, e.g., in covered areas, greenhouses or peculiar hilly regions and wheel odometry problems due to wheels slippage on sloped terrains, which is very typical in some crops such as vineyards. Unconstrained by external signals or references, visual odometry has been proven to be very significant by overcoming the limitations of other methodologies.

Keywords:

Precision agriculture; UAV; Remote sensing; Photogrammetry; Multispectral imaging; Density based clustering; Semantic interpretation; 3D point cloud segmentation; Real-time image processing; Agricultural field robots.

Table of Contents

1. Introduction

Agriculture plays a critical role in the economy of a given country and is considered the backbone of the economic system (Loizou et al., 2019). In the coming years, it will be essential to increase agriculture productivity by sustainable means to compensate the incessant increase of the population (Dethier & Effenberger, 2012), eventually requiring new innovative cost-effective technologies and enhanced solutions, aimed at making farming operations and processes more robust and economically feasible (Mahroof et al., 2021, Zaman et al., 2019).

In this context, Precision agriculture (PA) has been recognised as an essential approach to optimise essential crop-managing practices increasing field productivity and product quality while ensuring environmental protection/sustainability/safety (Ding et al., 2018; Grella et al., 2017; Lindblom et al., 2017). In large fields and in-fields located on hilly areas, the monitoring of crops and infield tasks may result in a laborious task, taking a lot of time and effort. The implementation of automatic machines and procedures could overcomes such criticalities (Comba et al., 2018; Grimstad and From, 2017). In this regard, robotics and automation play a crucial role, with particular reference to unmanned ground and aerial vehicles for timely crop monitoring and site-specific operations increasing productivity, e.g. in optimising fertilisers usage or precision weed control (Utstumo et al., 2018; Vakilian and Massah, 2017; De Baerdemaeker, 2013).

The autonomous vehicles to perform agricultural in-field tasks unsupervised or with least amount of external interaction it needs to be characterised by high level of automation (van Henten et al., 2013; Kassler, 2001). However, for autonomous machines navigation to perform operations, fully or even partially, within complex scenarios it is required to develop enhanced algorithms for effective path planning and management (Vidoni et al., 2015; Bechar & Vigneault, 2016). To achieve these conditions, an autonomous machine/vehicle requires the estimation of its instantaneous spatial position and the detailed spatial description of the surrounding area in which it is operating, e.g. inter-row width and crop canopy position and shape, to avoid damage/collision while performing the required task (Kassler et al., 2001; Van et al., 2013; Primicerio et al., 2015; Wang et al., 2019).

With recent advancement in remote sensing technologies, the use of three-dimensional path planning has led to enhanced performances resulting in collision free course from 3D obstacles along with advanced navigation strategies, overcoming the problems of standard 2D coverage (Han, 2018, Hameed, la Cour-Harbo, Osen, 2016). These advanced path planning strategies require the development of new 3D models, such as point clouds or triangulated meshes (Weiss & Biber, 2011; Miranda-Fuentes et al., 2015).

A raw 3D point cloud is a large dataset of points, representing the external visible surfaces of objects, in an arbitrary 3D coordinate system. A 3D point cloud dataset can be obtained using 3D sensors or by photogrammetry using structure from motion (SfM) software, processing appropriate sets of 2D images. In agricultural applications, numerous methodologies have been developed to obtain detailed 3D models of crop field from sensors optimised for 3D remote sensing, such as the light detection and ranging systems (LiDAR) (Mack et al., 2017) and by 3D depth cameras (Condotta et al., 2020).

The depth cameras widely utilised in agriculture can be divided into 3 major categories: stereoscopy (a camera with more than one lens with separate image sensors which allows the camera to imitate human binocular vision) (Luo et al., 2016), structured light (it is a 3D scanning device which utilises projected light patterns and a camera system to measure three dimensional shape of objects) (Saberioon & Cisar, 2016), and time-of-flight (a range imaging camera system utilising time of flight techniques) (Rosell-Polo et al., 2017; Bao, Tang, Srinivasan, Schnable, 2019). To generate 3D point clouds using SfM algorithms, a wide series of approaches have been explored by utilising images acquired by different cameras involving RGB, multispectral, hyperspectral and thermal sensors (Feng, Zhou, Vories, Sudduth, Zhang, 2020).

The substantial developments in UAVs and remote sensor technology have improved the acquisition quality of aerial imagery leading to the generation of highly accurate/detailed and dense 3D point clouds of crop field, cost effectively (Maes & Steppe, 2019; Wijesingha, Moeckel, Hensgen, Wachendorf, 2019). The 3D model representation of a field is opening the potential for new and improved scientific research and innovative precision agriculture solutions. However, new reliable tools such as advance algorithms to exploit 3D data in agriculture for detecting and mapping crops while identifying terrain and obstacles are needed (Mortensen et al., 2018; Comba et el., 2018). Also, these huge 3D data sets contain a massive amount of information that requires appropriate data extraction approaches and new processing algorithms, depending on the required final goal (Serazetdinova et al., 2019, Wolfert, Ge, Verdouw, Bogaardt, 2017; Van Evert et al., 2017; Pavón-Pulido et el., 2017; Zeybek & Şanlıoğlu, 2019).

Responding to the above discussion, research and development activities were carried out to supplement new innovative modelling framework to semantically interpret 3D point clouds of vineyards and to generate low complexity 3D mesh models of vine rows. In addition, to tackle the navigation and localisation issues due to complex agricultural scenarios, a cost-effective odometry system with enhanced image processing algorithm calibrated for the localisation and navigation of tracked vehicles on agricultural terrains was also developed.

1.1. Thesis content

Automated 3D path planning is a very important tool for automation and optimisation of robot operations in the field as it overcomes 2D path planning algorithms that ignore elevation changes of the vegetation, terrain, and obstacles (Hameed et al., 2016).

Since a 3D point cloud data in raw form is only a representation of points in geographic coordinates, in order to extract valuable information of the crop (e.g. crop rows distribution, shape and volume), the first step is to process and assign a label to each point of the 3D point cloud data, this task is also known as semantic interpretation/segmentation (Weinmann et al., 2015). Keeping in mind the first step, an innovative framework that processes 3D point clouds of vineyards in order to automatically identify, localise and cluster the individual vine rows has been developed. The proposed modelling is not hindered by complex scenarios, such as missing plants or non-linear vine rows, as it is able to automatically process non uniform vineyards. The **chapter 2** of the thesis explains in details all the steps involved in achieving this goal while considering all the problems imposed by unique characteristics of a vineyard.

Once each point of the 3D point cloud has been properly clustered, labelled and localised into canopy and terrain while filtering out extra vegetation, the next step was that of developing an innovative modelling framework in order to generate low complexity 3D mesh models of vine rows. The proposed methodology, based on a combination of convex hull filtration and minimum area c-gon design, significantly reduces the number of instances required to describe the spatial layout and shape of vine canopies allowing the amount of data to be drastically decreased (avoiding to compromise any relevant crop shape information such as volume, height of the canopy and inter-row space). This is a crucial task that allows shorter computational times for the processing of large datasets (e.g. raw 3D point clouds representing crops), thereby enabling in real time rapid communication and data exchange between in field robots/machines. The **chapter 3** of the thesis explains in detail the developed methodology which leads to the generation of low complexity 3D mesh models of vine rows.

In automated navigation and operations within a complex scenario, autonomous machines require proper knowledge of their up-to-date position and orientation assessment during movements (Ghaleb et al., 2017). To tackle navigation problems linked to limitations and drawbacks of GPS (e.g. poor/absent satellite signal) (Ericson and Åstrand, 2018) and wheel odometry (e.g. wheel slippage) (Bechar and Vigneault, 2016) a reliable and cost-effective monocular visual odometry system, calibrated for the localisation and navigation of tracked vehicles on agricultural terrains has been developed. The system is established on an enhanced image processing algorithm, founded on the cross-correlation approach. Unconstrained by external signals, the contribution of visual odometry in multi-source position control system has been proven to be very significant by overcoming the limitations of other methodologies (Scaramuzza and Fraundorfer, 2011). The **chapter 4** of the thesis presents in detail the developed visual odometry system and the enhanced algorithm.

1.2. References

1. Bechar, A., Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. Biosystems Engineering, 149, 94-111, 10.1016/j.biosystemseng.2016.06.014
2. Bao, Y., Tang, L., Srinivasan, S., Schnable, P.S. (2019). Field-based architectural traits characterisation of maize plant using time-of-flight 3D imaging. Biosystems Engineering, 178, 86-101, 10.1016/j.biosystemseng.2018.11.005
3. Condotta, I.C.F.S., Brown-Brandl, T.M., Pitla, S.K., Stinn, J.P., Silva-Miranda, K.O. (2020). Evaluation of low-cost depth cameras for agricultural applications. Computers and Electronics in Agriculture, 173, 105394, 10.1016/j.compag.2020.105394
4. Comba, L., Biglia, A., Ricauda Aimonino, D., Gay, P. (2018). Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. Computers and Electronics in Agriculture, 155, 84-95, 10.1016/j.compag.2018.10.005

5. Comba, L., Ricauda Aimonino, D., Gay, P. (2016). Robot ensembles for grafting herbaceous crops. Biosystems Engineering, 146, 227-239, 10.1016/j.biosystemseng.2016.02.012
6. De Baerdemaeker, J. (2013). Precision Agriculture Technology and Robotics for Good Agricultural Practices. IFAC Proceedings Volumes, 46, pp. 1-4, doi:10.3182/20130327-3-JP-3017.00003
7. Dethier, J-J., Effenberger, A. (2012). Agriculture and development: A brief review of the literature. Economic Systems, Elsevier, vol. 36(2), pages 175-205, 10.1016/j.ecosys.2011.09.003
8. Ding, Y., Wang, L., Li, Y., Li, D. (2018). Model predictive control and its application in agriculture: A review. Comput Electron Agr, 151, pp. 104-117, doi:10.1016/j.compag.2018.06.004
9. Ericson, S.K., Åstrand, B.S. (2018). Analysis of two visual odometry systems for use in an agricultural field environment. Biosystems Engineering, 166, pp. 116-125, doi:10.1016/j.biosystemseng.2017.11.009
10. Feng, A., Zhou, J., Vories, E.D., Sudduth, K.A., Zhang, M. (2020). Yield estimation in cotton using UAV-based multi-sensor imagery. Biosystems Engineering, 193, 101-114, 10.1016/j.biosystemseng.2020.02.014
11. Ghaleb, F.A., Zainala, A., Rassam, M.A., Abraham, A. (2017). Improved vehicle positioning algorithm using enhanced innovation-based adaptive Kalman filter. Pervasive and Mobile Computing, 40, pp. 139-155, doi:10.1016/j.pmcj.2017.06.008

12. Grimstad, L., From, P.J. (2017). Thorvald II - a Modular and Re-configurable Agricultural Robot. IFAC-PapersOnLine, 50, pp. 4588-4593, doi:10.1016/j.ifacol.2017.08.1005
13. Grella, M., Gil, E., Balsari, P., Marucco, P., Gallart M. (2017). Advances in developing a new test method to assess spray drift potential from air blast sprayers. Span J of Agric Res 15, doi:10.5424/sjar/2017153-10580
14. Hameed, I.A., la Cour-Harbo, A., Osen, O.L. (2016). Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. Robotics and Autonomous Systems, 76, 36-45, /10.1016/j.robot.2015.11.009
15. Han, J. (2018). An efficient approach to 3D path planning. Information Sciences, 478, 318-330, 10.1016/j.ins.2018.11.045
16. Kassler. M. (2001). Agricultural Automation in the new Millennium. Comput Electron Agr, 30, pp. 237-240, 10.1016/S0168-1699(00)00167-8
17. Loizou, E., Karelakis, C., Galanopoulos, K., Mattas, K. (2019). The role of agriculture as a development tool for a regional economy. Agricultural Systems, Volume 173, pp. 482-490, doi.org/10.1016/j.agsy.2019.04.002
18. Luo, L., Tang, Y., Zou, X., Ye, M., Feng, W., Li, G. (2016). Vision-based extraction of spatial information in grape clusters for harvesting robots. Biosystems Engineering, 151, 90-104, 10.1016/j.biosystemseng.2016.08.026
19. Lindblom, J., Lundström, C., Ljung, M., Jonsson, A. (2017). Promoting sustainable intensification in precision agriculture: review of decision

support systems development and strategies. Precis Agric, 18, pp. 309-331, doi:10.1007/s11119-016-9491-4

20. Mortensen, A.K., Bender, A., Whelan, B., Barbour, M.M., Sukkarieh, S., Karstoft, H., (2018). Segmentation of lettuce in coloured 3D point clouds for fresh weight estimation. Computers and Electronics in Agriculture, 154, 373-381, 10.1016/j.compag.2018.09.010
21. Miranda-Fuentes, A., Llorens, J., Gamarra-Diezma, J. L., Gil-Ribes, J. A., Gil, E. (2015). Towards an Optimized Method of Olive Tree Crown Volume Measurement. Sensors, 15(2), 3671-3687, 10.3390/s150203671
22. Mack, J., Lenz, C., Teutrine, J., Steinhage, V. (2017). High-precision 3D detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. Comput. Electron. Agric. 135, pp. 300–311, 10.1016/j.compag.2017.02.017
23. Mahroof, K., Omar, A., Rana, N. P., Sivarajah, U., Weerakkody, V. (2021). Drone as a Service (DaaS) in promoting cleaner agricultural production and Circular Economy for ethical Sustainable Supply Chain development. Journal of Cleaner Production, Volume 287, ISSN 0959-6526, 10.1016/j.jclepro.2020.125522
24. Maes, W.H., Steppe, K. (2019). Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture. Trends in Plant Science, 24, 152-164, 10.1016/j.tplants.2018.11.007
25. Primicerio, J., Gay, P., Ricauda Aimonino, D., Comba, L., Matese, A., Di Gennaro, S.F. (2015). NDVI based vigour maps production using automatic detection of vine rows in ultra-high resolution aerial

images. European Conference (10th) on Precision Agriculture, 465-470, DOI:10.3920/978-90-8686-814-8_57

26. Pavón-Pulido, N., López-Riquelme, J.A., Torres, R., Morais, R., Pastor, J.A. (2017). New trends in precision agriculture: a novel cloud-based system for enabling data storage and agricultural task planning and automation. Precision Agriculture, 18, 1038-1068, 10.1007/s11119-017-9532-7
27. Rosell-Polo, J.R., Gregorio, E., Gené, J., Llorens, J., Torrent, X., Arnó, J., et al. (2017). Kinect v2 sensor-based mobile terrestrial laser scanner for agricultural outdoor applications. IEEE/ASME Transactions on Mechatronics, 22, 2420-2427, 10.1109/TMECH.2017.2663436
28. Scaramuzza, D., Fraundorfer, F., (2011). Visual Odometry Part I: The First 30 Years and Fundamentals. IEEE Robotics & Automation Magazine, 18, pp. 80-92, doi:10.1109/MRA.2011.943233
29. Serazetdinova, L., Garratt, J., Baylis, A., Stergiadis, S., Collison, M., Davis, S. (2019). How should we turn data into decisions in AgriFood?. Journal of the Science of Food and Agriculture, 99, 3213-3219, 10.1002/jsfa.9545
30. Saberioon, M.M., Cisar, P. (2016). Automated multiple fish tracking in three-Dimension using a Structured Light Sensor. Computers and Electronics in Agriculture, 121, 215-221, 10.1016/j.compag.2015.12.014
31. Utstumo, T., Urdal, F., Brevik, A., Dørum, J., Netland, J., Overskeid, Ø. (2018). Robotic in-row weed control in vegetables. Comput Electron Agr, 154, pp. 36-45, doi:10.1016/j.compag.2018.08.043

32. Van Henten, E.J., Bac, C.W., Hemming, J., Edan, Y. (2013). Robotics in protected cultivation. IFAC Proceedings Volumes, 46, pp. 170-177, 10.3182/20130828-2-SF-3019.00070
33. Van Evert, F. K., Fountas, S., Jakovetic, D., Crnojevic V., Travlos, I., Kempenaar, C. (2017). Big Data for weed control and crop protection. Weed research, 57, 218-233, 10.1111/wre.12255
34. Vakilian, K.A., Massah, J. (2017). A farmer-assistant robot for nitrogen fertilizing management of greenhouse crops. Comput Electron Agr, 139, pp. 153-163, doi:10.1016/j.compag.2017.05.012
35. Vidoni, R., Bietresato, M., Gasparetto, A., Mazzetto, F. (2015). Evaluation and stability comparison of different vehicle configurations for robotic agricultural operations on side-slopes. Biosystems Engineering, 129, 197-211, 10.1016/j.biosystemseng.2014.10.003
36. Wijesingha, J., Moeckel, T., Hensgen, F., Wachendorf, M. (2019). Evaluation of 3D point cloud-based models for the prediction of grassland biomass. International Journal of Applied Earth Observation and Geoinformation, 78, 352-359, 10.1016/j.jag.2018.10.006
37. Wang, L., Wang, P., Liang, S., Qi, X., Li, L., Xu, L. (2019). Monitoring maize growth conditions by training a BP neural network with remotely sensed vegetation temperature condition index and leaf area index. Computers and Electronics in Agriculture, 160, pp. 82-90, 10.1016/j.compag.2019.03.017
38. Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant

features and efficient classifiers. ISPRS J Photogramm 105, 286-304, https://doi.org/10.1016/j.isprsjprs.2015.01.016

39. Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.J. (2017). Big Data in Smart Farming – A review. Agricultural Systems, 153, pp. 69-80, 10.1016/j.agsy.2017.01.023
40. Weiss, U., Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. Robotics and Autonomous Systems, 59, 265-273, 10.1016/j.robot.2011.02.011
41. Zeybek, M., Şanlıoğlu, İ. (2019). Point cloud filtering on UAV based point cloud. Measurement, 133, pp. 99-111, 10.1016/j.measurement.2018.10.013
42. Zaman, S., Comba, L., Biglia, A., Ricauda Aimonino, D., Barge, P., Gay, P. (2019). Cost-effective visual odometry system for vehicle motion control in agricultural environments. Computers and Electronics in Agriculture, 162, 82-94, 10.1016/j.compag.2019.03.037

2. Unsupervised localization of crop rows by enhanced density-based segmentation of 3D point cloud for precision agriculture

Abstract

The adoption of new sensors for crop monitoring and management usually leads to the acquisition of a large amount of data that require the development of specific algorithm for their processing. In this context, a 3D point cloud map of the crop, generated from remotely sensed data, can be of great importance. Valuable information extracted from 3D point clouds can be used, for example, for path planning of autonomous agricultural machines that can be thus adopted for in-field operations. However, since a 3D point cloud in its raw form is only a representation of the space (crop + surrounding area) in a 3D coordinate system, innovative algorithms that help in processing and analysing such complex 3D dataset are required.

This chapter presents an innovative segmentation method to automatically localise and cluster vine rows by processing 3D point clouds of vineyards. The algorithm provides as an output both an ordered set of 3D spatial coordinates representing the endpoints of each vine rows and a curve describing the spatial layout of the vine row. The proposed algorithm is also robust to the presence of curvilinear vine rows and/or missing plants. The useful information provided by the algorithm regarding the vine rows layout can be adopted for improving and optimising navigation and control strategies of autonomous agricultural machines that perform in-field operations.

Keywords: Precision agriculture; Remote sensing from UAV; Photogrammetry; Multispectral imaging; 3D data processing.

Nomenclature

$\mathcal{B}_{\mathrm{x,y}}$	cylindrical regions of the point cloud model $S_1^{\{\mathrm{Main}\}}$
$\mathcal{B}'_{\mathrm{x,y}}(\vartheta)$	slice of point cloud region model $\mathcal{B}_{\mathrm{x,y}}$
c	set of key points where $c \in \mathcal{V}_{\mathrm{m}}$
c_{i}	point closest to e_0 along the axis $x^{\{\mathrm{Loc_j}\}}$
c_{j}	set of key points canopy representing centre point of locally intersected vine rows
$c_{\mathrm{j}}^{\{\mathrm{Loc_b}\}}$	canopy centre points represented in a Local Cartesian Reference Frame $\{\mathrm{Loc_b}\}$
$c_{\mathrm{j}}^{\{\mathrm{Main}\}}$	canopy centre points represented in a Local Main Reference Frame $\{\mathrm{Main}\}$
$\mathcal{C}$	comprehensive set of canopy centre points
d	ellipsoid vertical radius
$d_v(x, y)$	map of local inter row spacing
d_v	local inter row width
D_{y}	point cloud density of points p_{i} along the $\mathrm{y}^{\{\mathrm{Loc}\}}$ axis
$D_{\mathrm{y}}(\mathcal{B}'_{\mathrm{x,y}}, s)$	normalised frequencies distribution histogram of points $p_{\mathrm{i}} \in \mathcal{B}'_{\mathrm{x,y}}$ along the $\mathrm{y}^{\{\mathrm{Loc}\}}$ axis
e	ordered set of enhanced key points
e_0	the first enhanced key point
$e^{[\mathrm{h}]}$	enhanced key point defined at iteration $[\mathrm{h}]$
e_-	closest point c_{i} along the negative axis $x^{\{\mathrm{Loc_j}\}}$
e_+	closest point c_{i} along the positive axis $x^{\{\mathrm{Loc_j}\}}$
$[\mathrm{h}]$	algorithm iteration
K	points enclosed within a circle having a radius
l_{G}	grid step
$n_{\mathcal{C}}$	overall number of detected key points $\mathcal{C}$
$\mathrm{n}_{\mathcal{V}}$	overall number of detected vine rows $\mathcal{V}_{\mathrm{m}}$

$O_{\mathrm{Loc}}^{\{\mathrm{Main}\}}$	origin of local reference frame {Loc} in Main reference frame
p_{i}	points belonging to the slice of point cloud model $\mathcal{B}_{\mathrm{x,y}}$
$r_{\mathcal{B}}$	radius of the cylindrical subset $\mathcal{B}_{\mathrm{x,y}}$
$R_{\mathrm{Main}}^{\mathrm{Loc}}$	rotation matrix from {Loc} to {Main} reference frame
$S_{1}^{\{\mathrm{Main}\}}$	3D point cloud model in {Main} reference frame
s_{y}	bin of histogram S_{y}
S_{y}	set of all the histogram bins
t	distance threshold between enhanced key points ~~e_{u} and c_{i}~~
U	set of refined key points
$\hat{v}_{\mathrm{j}}$	local maxima in the density histogram $D_{\mathrm{y}}(\mathcal{B}'_{\mathrm{x,y}}, s)$
$\mathcal{V}_{\mathrm{m}}$	group of set of points representing individual vine rows
$\mathcal{V}_{\mathrm{m}}^{[\mathrm{h}-1]}$	vine row cluster defined at iteration $[\mathrm{h}-1]$
$\mathcal{V}_{\mathrm{m}+1}^{[\mathrm{h}]}$	vine row cluster defined at iteration $[\mathrm{h}]$
$x_{\mathrm{b}}, y_{\mathrm{b}}$	cylindrical region sample $\mathcal{B}_{\mathrm{x_b,y_b}}$ centre location
$\mathrm{xy}^{\{\mathrm{Main}\}}$	axis of Main reference frame
$\mathrm{x}^{\{\mathrm{Loc}\}}$ & $\mathrm{y}^{\{\mathrm{Loc}\}}$	x and y axis of the {LOC}, parallel and perpendicular to the vine row
z_{b}	local elevation of the digital terrain model
$\mathrm{z}^{\{\mathrm{Main}\}}$	axis of Main reference frame

Greek letters

ϑ_{v}	local vine row orientation
$\vartheta_{v}(x,y)$	map of local vine row direction
$\vartheta_{\perp v}$	angle perpendicular to the local vine row orientation, measured anticlockwise from the horizontal $\mathrm{x}^{\{\mathrm{Main}\}}$ axis
$\vartheta_{v}(x,y)$	local vine row direction angle
δ_{s}	bin width of histogram D_{y}

$\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$	neighbourhood of points c_{j} within an elliptic Region of Interest at iteration $[\mathrm{h}]$

Acronyms

2D	two dimensional
3D	three dimensional
DTM	digital terrain model
$\{\mathrm{Loc}\}$	local metrical reference frame
$\{\mathrm{Main}\}$	main reference frame
ROI	region of interest

2.1 Introduction

Unmanned ground and aerial vehicles (UGVs and UAVs, also named drones) are assuming a key role in modern farming known as Agriculture 4.0 (Rao Mogili et al., 2018; Michels et al., 2020). Indeed, drones capability to autonomously perform in-field operations such as seeds distribution, data acquisition, fertilizers and pesticides spraying are being profitably exploited in many agricultural scenarios (Kerkech et al., 2020; Peng and Vougioukas, 2020; Thompson and Puntel, 2020). The agricultural tasks that benefit or might benefit by the adoption of autonomous ground and aerial drones can be grouped into two main categories: crop monitoring and in-field operations. Remote sensing and proximal/close range sensing by UGVs and UAVs have already proved their effectiveness in many applications, such as canopy vigour assessment (Campos et al., 2019; Khaliq et al., 2019), nitrogen estimation (Colorado et al., 2020), plants high-throughput phenotyping traits evaluation (Sun et al, 2020; Xie and Yang, 2020), crop mapping (Primicerio et al. 2017; Mazzia et al., 2020) or disease detection (Kerkech et al., 2020). For what concern in-field operations, valuable solutions based on robotic drones involve transplanting and seedling (Nagasaka et al., 2009), pruning and thinning (Zahid et al., 2020), weed control (McAllister et al., 2019) and harvesting (Bechar et al., 2017).

The effectiveness of the adoption of drones for precision agriculture applications is strictly related to the proper knowledge of the working environment, in terms of both spatial layout (Chen et al., 2020; Gao et al., 2020) and crops status. Indeed, the joint contribution of such information allows to timely reach the target (or to properly modulate the agronomical operation) and ensure, at the same time, the safe accomplishment of the operation (Gil et al., 2013; Wang et al., 2020). In this context, accurate and reliable path planning and control strategies of drones are thus essential to properly achieve the tasks (Dusadeerungsikul and Nof, 2019; Khajepour et al., 2020). To this aim, the spatial description of the environment in which the autonomous vehicles have to move and operate is required (Graf Plessen and Bemporad, 2017; Li et al., 2020).

Sensors able to provide 3D models of the agricultural environment can lead to favourable improvement in the description of complex scenarios in which drones operate (Chakraborty et al., 2019; Comba et al., 2019; Zhang et al., 2020). Some examples of enhanced spatial information derived from 3D models regard fruit position for automatic harvesting (Kang and Chen, 2020; Wu et al., 2020), canopy shape and size for variable spraying (Llorens et al., 2011; Grella et al., 2020), branches location for automatic pruning (Cuevas-Velasquez et al., 2020), and crop location for accurate path planning (Sanz et al., 2018; Jurado et al., 2020). Such 3D models are usually in the form of point cloud, which is a set of unordered points in the 3D space. A 3D point cloud can be derived by Structure from

Motion (SfM) algorithms (Gené-Mola et al., 2020), light detection and ranging systems (LiDAR) (Blanquart et al., 2020; Shendryk et al., 2020) or depth cameras (Condotta et al., 2020). However, specific algorithms have to be developed to properly extract valuable information from raw 3D models (Escolà et al., 2017; Comba et al., 2020a), even based on recent artificial intelligence tools (Zhang et al., 2021). A crucial phase of processing algorithms is usually the semantic segmentation of 3D point clouds, which assigns each point to different portions of the whole model, such as leaves, branches, fruits and other elements (Mortensen et al., 2018; Zhou et al., 2019; Zeng et al., 2020; Comba et al., 2020b). However, in order to fully automate the 3D point cloud processing, the automatic detection of the crop (e.g. row, plant, trees, etc.) from the 3D model of the considered agricultural environment is usually required (Matese et al., 2019; Comba et al., 2020a; Comba et al., 2020b). This is a crucial phase in the interpretation of complex and huge 3D point cloud of agricultural environments, moving from a macro level (parcel and plot scale) to a micro level (plants, fruits, branches).

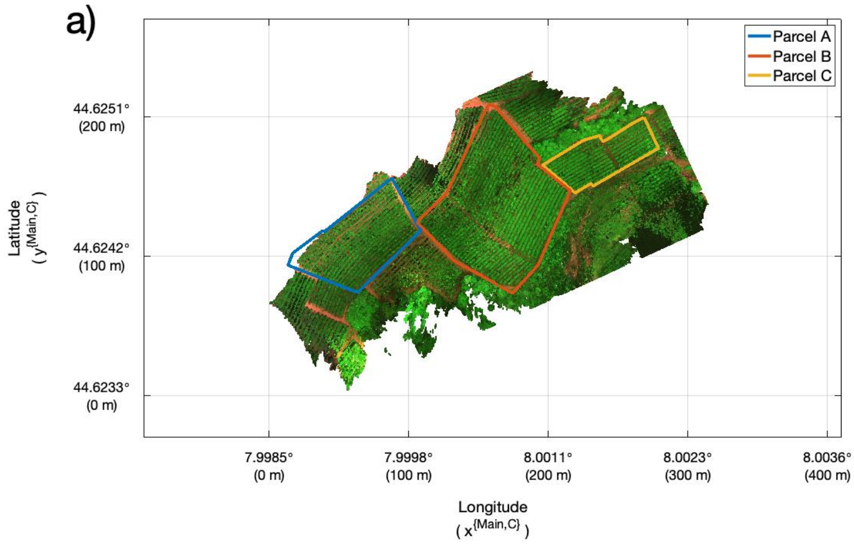
In this chapter an innovative segmentation algorithm to automatically localise and cluster vine rows by processing 3D point clouds of vineyards is described. The algorithm provides as an output both an ordered set of 3D spatial coordinates representing the two ends position of each vine rows and a curve describing the spatial layout of the vine row. Peculiarities of vineyard scenarios, such as curvilinear vine rows, missing plants or diseased vines (which are reflected in the 3D points clouds of the region), require specific solutions and prevent the adoption of already available methodologies (e.g. Ester et al., 1996 in Matlab®; Weinmann et al., 2015). In addition, the information provided by the proposed algorithm can be exploited in automated 3D path planning, which is a key task for the automation and optimisation of drone operations in the field. Indeed, overcoming 2D path planning algorithms, 3D path planning fully exploits terrain and environment characteristics (Jin and Tang, 2011; Hameed et al., 2016).

The chapter structure is as follows: Section 2.2 describes the experimental field and the acquisition campaigns are, Section 2.3 presents the innovative algorithm for the vine rows localisation and clustering, results are presented and discussed in Section 2.4, while the conclusions are reported in Section 2.5.

2.2 Case study and data acquisition

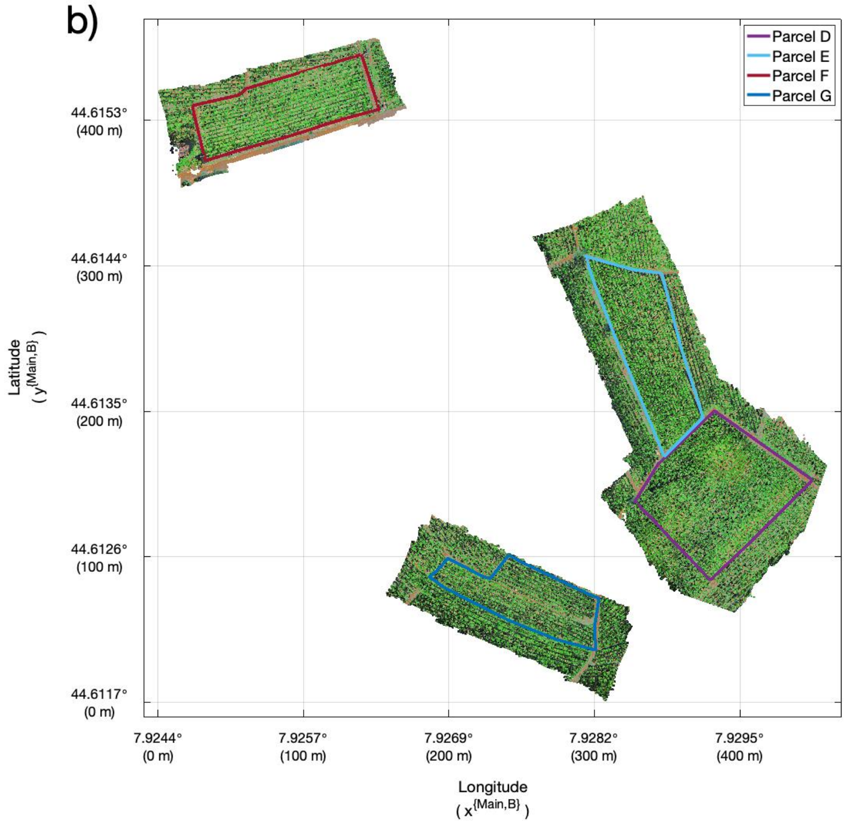
In this work, a set of seven parcels, three of which located in Serralunga d'Alba and four in Barolo (Piedmont, Northwest of Italy), was considered as case study. The seven parcels covered an overall surface of about 3.4 hectares, and they were characterised by a sloped land conformation. Six parcels (A, B, D, E, F, and G, see Fig. 2.1) were cultivated with Nebbiolo vine variety and one parcel (C, see Fig. 2.1) was cultivated with Moscato vine variety using a vertical shoot position trellis system. The space between vine plants and the inter-row space were about 0.9 meter and 2.5 meter, respectively.

a)



Parcel A
Parcel B
Parcel C
44.6251°
(200 m)
44.6242°
(100 m)
44.6233°
(0 m)
Latitude
(y{Main,C})
7.9985°
(0 m)
7.9998°
(100 m)
8.0011°
(200 m)
8.0023°
(300 m)
8.0036°
(400 m)
Longitude
(x{Main,C})

b)



Parcel D
Parcel E
Parcel F
Parcel G
44.6153°
(400 m)
44.6144°
(300 m)
44.6135°
(200 m)
44.6126°
(100 m)
44.6117°
(0 m)
Latitude
(y{Main,B})
7.9244°
(0 m)
7.9257°
(100 m)
7.9269°
(200 m)
7.9282°
(300 m)
7.9295°
(400 m)
Longitude
(x{Main,B})

Fig. 2.1. 3D point clouds of the seven parcels considered as case study and located in Serralunga d'Alba - Italy (a) and in Barolo - Italy (b). RGB chromatic coordinates of the 3D point clouds were only adopted for graphical purposes but not required to run the developed algorithm.

The 3D point clouds were obtained by using Agisoft Photoscan® software (2020, St. Petersburg, Russia), which is based on structure from motion (SfM) algorithms to process UAV-based aerial images. Structure from motion photogrammetry approach approximates a 3D structure using 2D images. A Parrot Sequoia® multispectral camera (Parrot©, 2018, Paris, France) was used to acquire the aerial images with a resolution of 1280 × 960 pixels. The UAV flights were done in Serralunga d'Alba and in Barolo in 2018 and 2019 respectively. The height of the UAV flight was maintained close to 35 m with respect to the terrain by using a set of waypoints, which were defined on the basis of the vineyard geographic information system map. A forward and side overlap greater than 80% was guaranteed between adjacent images, this helps in the images alignment process. Prior to the images alignment, a radiometric calibration was performed on the images by using the reference images of a Micasense calibrated reflectance panel (Seattle, Washington, USA), which were acquired before and after each UAV flight.

It should be noted that the obtained 3D point clouds of vineyards had neither colour nor spectral information, so that only spatial information provided by each point of the clouds was exploited by the proposed processing method. Each point of the cloud may help in defining the shape or identifying different objects such as vines, inter row paths, or other elements typical of a vineyard in a 3D coordinate system. Point clouds are indeed a means of collating a large number of single spatial measurements into a dataset that can then represent a whole. However, the proposed processing method is affected neither by the type of airborne sensor nor by the spectral difference that could characterise different vineyard environments during the growing season.

2.3. New 3D point cloud processing method

The developed algorithm, which automatically localises / detects and clusters vine rows within a 3D point cloud, has two main outputs: (i) an ordered set of points in a 3D coordinate system representing the two ends position of each vine row and (ii) a curve representing the spatial layout of each row. The curve, which is tangent to the vine row, is the projection of the centres of vines canopy on the terrain surface. The vine rows detected within the 3D point cloud are automatically sorted and numbered, and the algorithm also provides their length, difference of altitude and orientation with respect to the west-east direction.

The algorithm can be summarised into three phases: (1) the detection of a set of key points representing the centre of vines canopy by using a semantic segmentation approach, (2) the clustering of the key points to identify the single vine row (please note that the raw 3D point cloud and the detected key points are an unordered set of points in 3D coordinates), and, finally, (3) the sorting and refinement procedure applied to each cluster of key points to determine the curves that characterises the location of each vine row.

Specific criticalities that characterise a 3D point cloud of vineyards, which prevent the adoption of already available processing algorithms, have been discussed in detail in each processing phase, together with the innovative solutions that have been defined.

2.3.1 Key points detection

The first phase of the algorithm is the detection of a set of key points $c_{\mathrm{j}} = \left[x_{\mathrm{j}}\ y_{\mathrm{j}}\ z_{\mathrm{j}}\right]^{\mathrm{T}}$, representing the canopy central points of the vine rows, which is obtained by processing the raw 3D point cloud of the vineyard. This first step allows the main information of the canopy, required in the second and third phases, to be extracted from the raw 3D point cloud. Positions of the key points were obtained by modifying and updating a version of an algorithm of the authors presented in Comba et al. (2019).

In Comba et al. (2019), in order to identify the vineyard parcels, the raw 3D point cloud ($S_1^{\{\mathrm{Main}\}}$) was scanned by slicing the cloud with a mobile window and selecting different cylindrical regions (named $\mathcal{B}_{\mathrm{x,y}}$), centred in $[x\ y]^{\mathrm{T}}$ and with radius $r_{\mathcal{B}}$, within the sliced 3D point cloud. A vineyard likelihood test was performed on each cylindrical region $\mathcal{B}_{\mathrm{x,y}}$ by selecting slices, named $\mathcal{B}'_{\mathrm{x,y}}(\vartheta)$, from the cylindrical region $\mathcal{B}_{\mathrm{x,y}}$ (see Fig. 2.2a). An example of a cylindrical region $\mathcal{B}_{\mathrm{x_b,y_b}}$, centred in $[x_{\mathrm{b}}, y_{\mathrm{b}}]^{\mathrm{T}} = [200{,}130]^{\mathrm{T}}$ m with a radius $r_{\mathcal{B}} = 5$ m, is represented in Fig. 2.2b (green dots) together with the subset $\mathcal{B}'_{\mathrm{x_b,y_b}}(\vartheta_{\perp v})$, which was selected using the angle value $\vartheta_{\perp v} = 1.1\pi$ (red dots). The angle ϑ, provided by the method of Comba et al. (2019), identifies the perpendicular direction with respect to the local vine row orientation. The angle ϑ is measured anticlockwise from the $\mathrm{x}^{\{\mathrm{Main}\}}$ axis on the horizontal $\mathrm{xy}^{\{\mathrm{Main}\}}$ plane. For the complete discussion about this procedure, please refers to Comba et al. (2019).

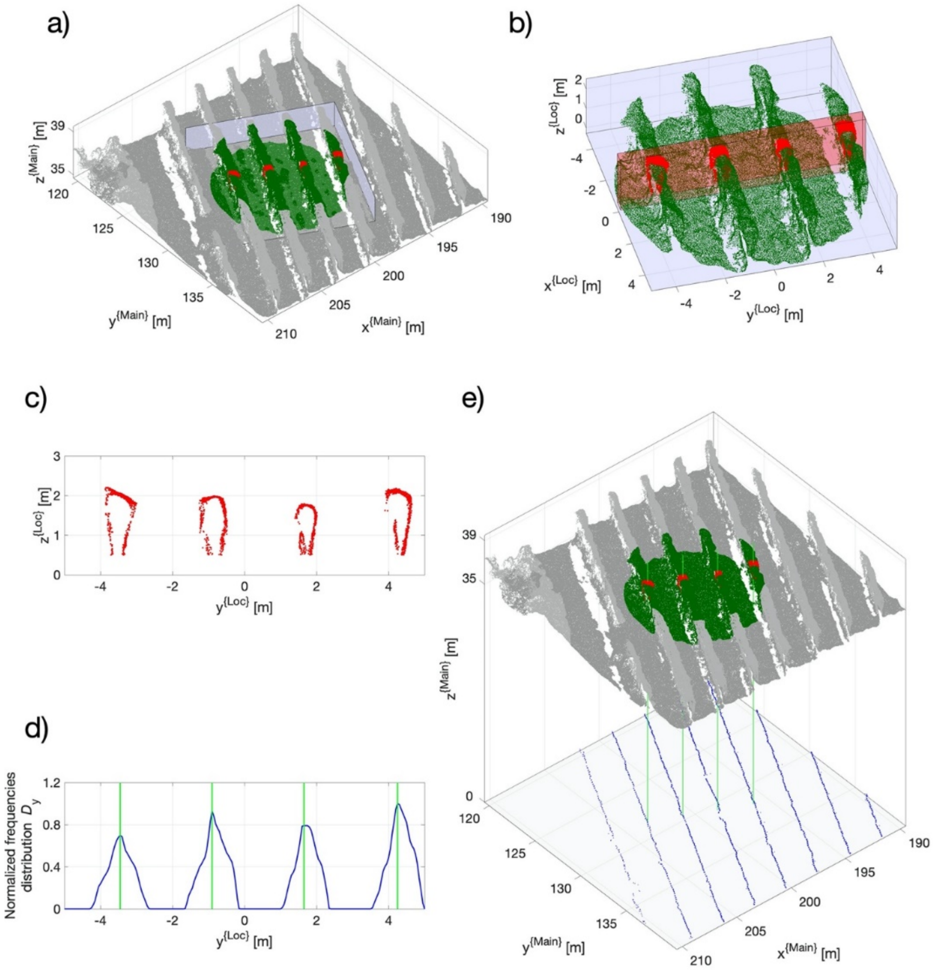


Figure 2.2. (a) Cylindrical subset $\mathcal{B}_{\mathrm{x_b,y_b}}^{\{\mathrm{Main}\}}$ (green dots) of the 3D point cloud $S_1^{\{\mathrm{Main}\}}$ (grey dots) represented in the Local Reference Frame $\{\mathrm{Loc}\}$, with origin in $O_{\mathrm{Loc}}^{\{\mathrm{Main}\}} = (x_\mathrm{b}, y_\mathrm{b}, z_\mathrm{b}) = (200, 130, 35)$ m; (b) enlargement of the subset $\mathcal{B}_{\mathrm{x_b,y_b}}^{\{\mathrm{Loc}\}}$ (green dots) and $\mathcal{B}'_{\mathrm{x_b,y_b}}(\vartheta_{\perp v})$ (red dots), with $\vartheta_{\perp v} = 1.1\pi$; (c) projection of the subset $\mathcal{B}'_{\mathrm{x_b,y_b}}(\vartheta_{\perp v})$ on the $\mathrm{yz}^{\{\mathrm{Loc}\}}$ plane; (d) normalised frequencies distribution histogram $D_\mathrm{y}(\mathcal{B}'_{\mathrm{x_b,y_b}}, s)$ (blue line) with the detected local maxima $\hat{v}_\mathrm{j}$ (green line);

(e) detected central points $c_{\mathrm{j}}^{\{\mathrm{Main}\}}$(green line) of the canopy in the original 3D point cloud $S_1^{\{\mathrm{Main}\}}$ (grey dots) and entire set central points $\mathcal{C}$ (blue dots) of the canopy.

In this work, the points density distribution within the subset $\mathcal{B}'_{\mathrm{x,y}}(\vartheta_{\perp v})$ (which for definition intersects several vine rows being $\vartheta_{\perp v}$ the angle defining the perpendicular direction with respect to the local vine row orientation) was exploited to determine the position of the central points c_{j} of the interested canopy vine rows. To perform this task, a Local Reference Frame $\{\mathrm{Loc_b}\}$ was introduced, which has the origin located in $O_{\mathrm{Loc_b}}^{\{\mathrm{Main}\}} = [x_{\mathrm{b}}, y_{\mathrm{b}}, z_{\mathrm{b}}]^{\mathrm{T}}$ and with $\mathrm{x}^{\{\mathrm{Loc_b}\}}$ and $\mathrm{y}^{\{\mathrm{Loc_b}\}}$ axis parallel and perpendicular to the vine row respectively (see Figs. 2.2a and 2.2b). The coordinate z_{b} was chosen equal to the local elevation of the digital terrain model (DTM). The density of the 3D point cloud was thus assessed by computing the normalised frequencies distribution histogram of points p_{i} along the $\mathrm{y}^{\{\mathrm{Loc_b}\}}$ axis

$$D_{\mathrm{y}}(\mathcal{B}'_{\mathrm{x,y}}, s) \ = \ \mathrm{card}\{p_{\mathrm{i}} = [x, y, z]^{\mathrm{T}} \in \mathcal{B}'_{\mathrm{x,y}}(\vartheta_{\perp v}) \colon |\mathrm{y} - s_{\mathrm{y}}| < \frac{\delta_{\mathrm{s}}}{2}\} \cdot \mathrm{card}\left(\mathcal{B}'_{\mathrm{x,y}}(\vartheta_{\perp v})\right)^{-1} \tag{1}$$

with $s_{\mathrm{y}} \in S_{\mathrm{y}} = \{-r_{\mathcal{B}}, -r_{\mathcal{B}} + \delta_{\mathrm{s}}, -r_{\mathcal{B}} + 2\delta_{\mathrm{s}}, \dots, 0, \dots, r_{\mathcal{B}}\}$, where S_{y} is the set of all the histogram bins and δ_{s} is bins width. The normalised frequencies distribution histogram obtained by processing the sample $\mathcal{B}'_{\mathrm{x_b,y_b}}(\vartheta_{\perp v})$ is reported in Fig. 2.2d. More in detail, the position of $c_{\mathrm{j}}^{\{\mathrm{Loc_b}\}}$ was determined by detecting the local maxima $\hat{v}_{\mathrm{j}}$ in the density histogram $D_{\mathrm{y}}(\mathcal{B}'_{\mathrm{x,y}}, s)$, as:

$$c_{\mathrm{j}}^{\{\mathrm{Loc_b}\}} = \left[0, \hat{v}_{\mathrm{j}}, 0\right]^{\mathrm{T}}. \tag{2}$$

The central points $c_{\mathrm{j}}^{\{\mathrm{Loc_b}\}}$ of the canopy are thus represented in a Local Cartesian Reference Frame $\{\mathrm{Loc_b}\}$, and their absolute position in the Main Reference Frame $\{\mathrm{Main}\}$ was reconverted as follows

$$\begin{aligned} c_{\mathrm{j}}^{\{\mathrm{Main}\}} &= \begin{bmatrix} x_{\mathrm{j}} \\ y_{\mathrm{j}} \\ z_{\mathrm{j}} \end{bmatrix} = R_{\mathrm{Main}}^{\mathrm{Loc}} \cdot c_{\mathrm{j}}^{\{\mathrm{Loc_b}\}} + O_{\mathrm{Loc}}^{\{\mathrm{Main}\}} \\ &= \begin{bmatrix} \cos\vartheta_{\perp v} & -\sin\vartheta_{\perp v} & 0 \\ \sin\vartheta_{\perp v} & \cos\vartheta_{\perp v} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \hat{v}_{\mathrm{j}} \\ 0 \end{bmatrix} + \begin{bmatrix} x_{\mathrm{b}} \\ y_{\mathrm{b}} \\ z_{\mathrm{b}} \end{bmatrix} \end{aligned} \tag{3}$$

where $R_{\mathrm{Main}}^{\mathrm{Loc}}$ is the rotation matrix from $\{\mathrm{Loc}\}$ to $\{\mathrm{Main}\}$ and $O_{\mathrm{Loc}}^{\{\mathrm{Main}\}}$ is the $\{\mathrm{Loc}\}$ origin, expressed in the $\{\mathrm{Main}\}$ coordinates. For example, the detected local maxima $\hat{v}_{\mathrm{j}}$ in the density histogram $D_{\mathrm{y}}(\mathcal{B}'_{\mathrm{x,y}}, s)$ are reported in green in Fig. 2.2d, while the detected vine row centres $c_{\mathrm{j}}^{\{\mathrm{Main}\}}$ are green highlighted in Fig. 2.2e. In the example of Fig. 2.2, four local maxima $\hat{v}_1 = -3.45$, $\hat{v}_2 = -0.90$, $\hat{v}_3 = 1.65$ and $\hat{v}_4 = 4.25$ were detected that led to the four canopy central points $c_1^{\{\mathrm{Main}\}} = [203.13, 128.55, 35.03]^{\mathrm{T}}$, $c_2^{\{\mathrm{Main}\}} = [200.82, 129.62, 35.08]^{\mathrm{T}}$, $c_3^{\{\mathrm{Main}\}} = [198.50, 130.69, 35.09]^{\mathrm{T}}$ and $c_4^{\{\mathrm{Main}\}} = [196.14, 131.78, 35.07]^{\mathrm{T}}$ meters.

The procedure described in the previous paragraphs has to be applied to the entire 3D point cloud to obtain a set of central points $\mathcal{C} = \left\{c_{\mathrm{j}}, \mathrm{j} = 1, \dots, n_{\mathcal{C}}\right\}$ of the canopy of each vine rows, where $n_{\mathcal{C}}$ is the overall number of detected key points. Performing this modified scouting procedure on the set of subsets $\mathcal{B}_{\mathrm{x_k,y_k}}$, with $[x_{\mathrm{k}}, y_{\mathrm{k}}] \in G$, covering thus the entire point-cloud map S, a comprehensive set of canopy centre points $\mathcal{C} = \left\{c_{\mathrm{j}}, \mathrm{j} = 1, \dots, n_{\mathcal{C}}\right\}$ can be obtained, where $n_{\mathcal{C}}$ is the overall number of detected key points. The grid step l_{G}, used for analysing the 3D point cloud slice by slice, was considered equal to 0.5 m, which is a trade-off between a good spatial resolution of the results and the computational time. To simplify the discussion of the next phases of the algorithm, the projection of the key points $\mathcal{C}$ on the 2D plane $z^{\{\mathrm{Main}\}} = 0$ will be considered as shown in Fig. 2.2e. All the results will be easily reported in the original 3D system by restoring the DTM, considering the local terrain elevation as z coordinate of points. Results obtained by processing the entire 3D point cloud S_1 of Fig. 2.1a are reported in Fig. 2.3.

The obtained set of central points $\mathcal{C}$ of the canopy, together with additional information regarding vineyards local features provided by Comba et al. (2019) (the maps of local vine row direction $\vartheta_v(x, y)$ and local inter row spacing $d_v(x, y)$) will be used in the following algorithm phases.

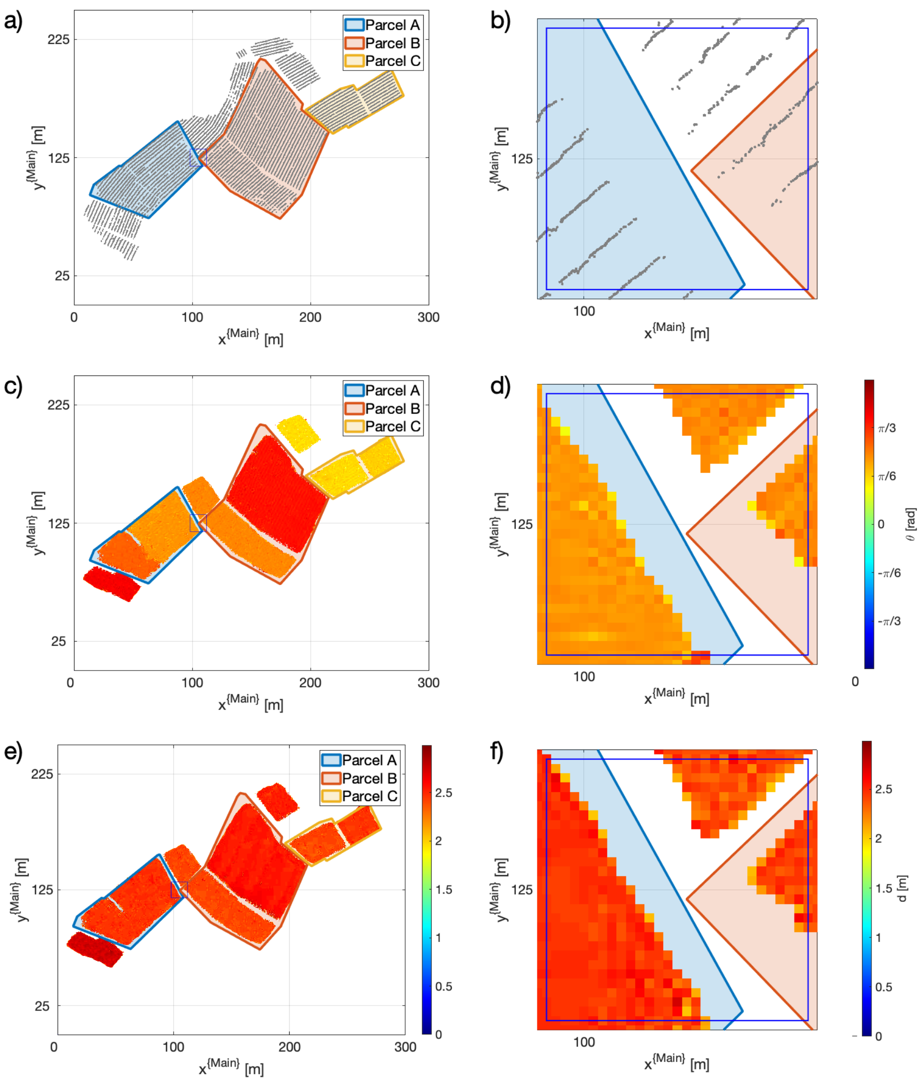


Figure 2.3. (a) the comprehensive set of canopy centre points $\mathcal{C}$, (c) the map of local vine row direction $\vartheta_v(x,y)$ obtained using the algorithm explained in Comba et al. (XX) and (e) the map of local inter row spacing $d_v(x,y)$, obtained by processing the whole point cloud S_1 of Fig. 2.1a, and their enlargement (b), (d) and (f), respectively.

757

2.3.2 Density-based key points clustering

The obtained set $\mathcal{C}$ of central points of the canopy of all the vine rows is unordered and unclustered. To detect the location of each vine row, a clustering and sorting procedure is thus required. The output of the second phase of the algorithm is indeed a set of clusters $\mathcal{V}_{\mathrm{m}}$ of points, with $\mathrm{m} = 1, \ldots, \mathrm{n}_{\mathcal{V}}$ where $\mathrm{n}_{\mathcal{V}}$ is the number of the detected vine rows. It should be noted that points within the set $\mathcal{C}$, representing a single vine row, are characterised by a particular spatial layout that has a predominant dimension (vine row length) with respect to the inter group distance (here represented by the inter-row path width). In addition, the occurrence of missing and/or diseased plants lead to vine row interruptions that, in some cases, can be more extended than the inter-row path. Indeed, with an inter-row path ranges between 2 and 3 m and with a vine plants distance ranging from 0.9 to 1.2 m, as shown in Fig. 4a, the occurrence of two consecutive missing and/or diseased plants represents an obstacle to clustering; this involves all the available 3D point cloud clustering methods to fail.

The proposed clustering density-based approach performs an iterative clustering task defining small subsets of points representing sections of vine row and merging them sequentially when specific criteria are fulfilled. Considering one algorithm iteration $[\mathrm{h}]$, first, a neighbourhood $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ of point c_{j} within an elliptic region of interest (ROI) was defined as

$$\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]} = \left\{ [x, y, z]^T \in \mathcal{C}^{\{\mathrm{Main}\}} \;\middle|\; \begin{array}{c} \dfrac{\left(x \cdot \cos\vartheta_v + y \cdot \sin\vartheta_v - x_{\mathrm{j}}\right)^2}{3 \cdot d_v} + \\ \dfrac{\left(-x \cdot \sin\vartheta_v + y \cdot \cos\vartheta_v - y_{\mathrm{j}}\right)^2}{0.5 \cdot d_v} \leq 1 \end{array} \right\} \tag{4}$$

where d_v is the local inter-row width [m], ϑ_v is local vine row orientation (Comba et al. (2019)), and $(x_\mathrm{j}, y_\mathrm{j})$ are the coordinates of the key point c_j. In the sample dataset used in the algorithm description, values d_v and ϑ_v can be derived from specific map, such reported in Fig. 2.3b and Fig. 2.3c. Equivalently, considering the Local Reference Frame $\{\mathrm{Loc_j}\}$ defined with the origin in c_j and with axis $x^{\{\mathrm{Loc_j}\}}$ and $y^{\{\mathrm{Loc_j}\}}$ tangent and perpendicular to the local vine row orientation ϑ_v, respectively (similarly to the previous processing step), neighbourhood $\mathcal{E}_\mathrm{j}^{[\mathrm{h}]}$ of point c_j can be more briefly expressed as

$$\mathcal{E}_\mathrm{j}^{[\mathrm{h}]} = \left\{ [x, y, z]^T \in \mathcal{C}^{\{\mathrm{Loc_j}\}} \,\middle|\, \frac{x^2}{3 \cdot d_v} + \frac{y^2}{0.5 \cdot d_v} \leq 1 \right\} \quad (5)$$

An example of selected neighbouring points $\mathcal{E}_\mathrm{j}^{[\mathrm{h}]}$ of c_j within the elliptic ROI is represented in Fig. 2.4a, while its enlargement is reported in Fig. 2.4b.

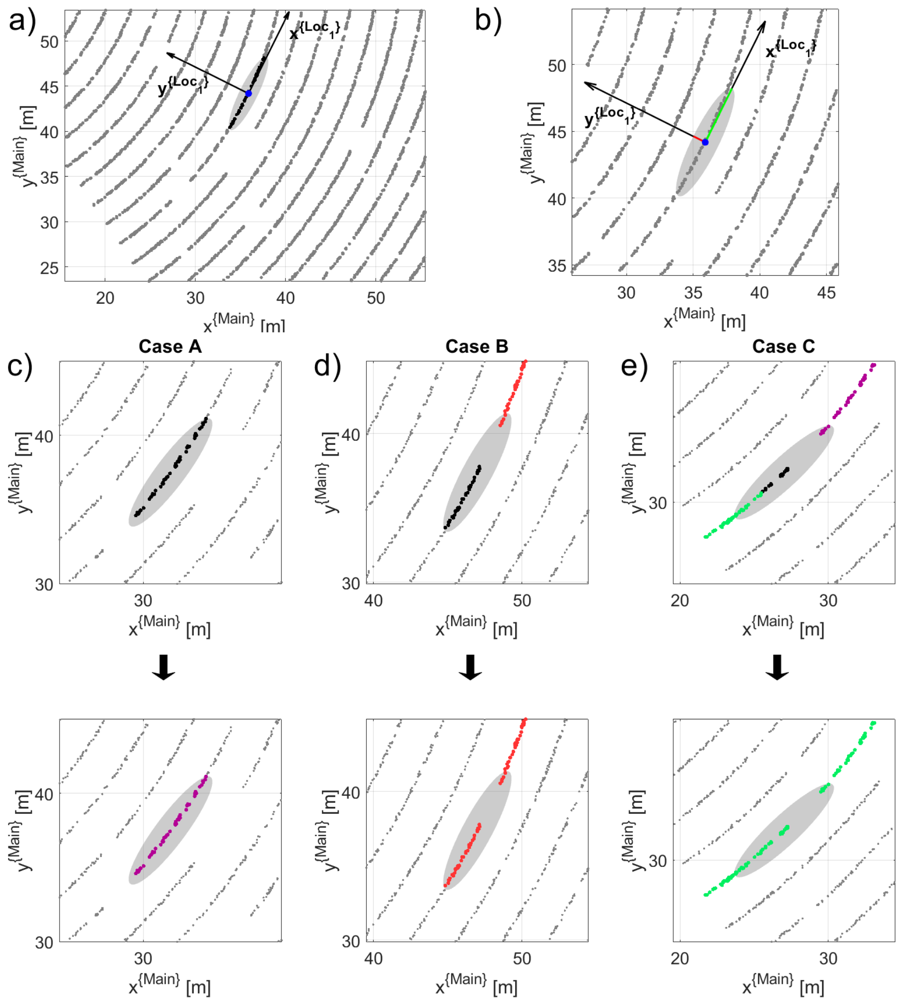


Figure 2.4. (a) the entire set of central points $\mathcal{C}$ (grey dots) of the canopy in the Local Reference Frame $\{\mathrm{Loc_j}\}$, and an example of the elliptic ROI (grey area) that is used to select the neighbouring points $\mathcal{E}_\mathrm{j}$ (black dots) of c_j, located in $[35.90\ 44.18]$ (blue dot); (b) enlargement of subplot (a) to highlight the semi axis of the ROI (major in green line and minor in red line); (c) Case A scenario of the clustering process where points $\mathcal{E}_\mathrm{j}^{[\mathrm{h}]}$ do not belong to any defined vine row cluster $\mathcal{V}_\mathrm{m}^{[\mathrm{h-1}]}$,

points $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ are thus defined as a new cluster $\mathcal{V}_{\mathrm{m+1}}^{[\mathrm{h}]} = \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$; (d) Case B scenario of the clustering process where points within the subset $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ belong to an already defined cluster $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$, points $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ and $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ are merged forming the updated cluster $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h}]} = \mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]} \cup \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$; (e) Case C scenario of the clustering process where points within $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ belong to two different clusters $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ and $\mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]}$, all points are merged together in the oldest one, thus updating $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ as $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h}]} = \mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]} \cup \mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]} \cup \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ and deleting the cluster $\mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]}$.

Then, depending on the status of points $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$, three scenarios of cluster assignment can occur. First (case A), when points $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ do not belong to any previously defined vine row cluster $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$, a new cluster $\mathcal{V}_{\mathrm{m+1}}^{[\mathrm{h}]} = \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ is defined. Second (case B), at least one point within subset $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ belongs to an already defined cluster $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ and, thus, $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ and $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ are merged forming the updated cluster $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h}]} = \mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]} \cup \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$. Third (case C), it is an extension of case B and it occurs when some (or even all) points within $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ belong to more than one cluster, such as $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ and $\mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]}$. In this case, all the clusters are merged in the oldest one, together with $\mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$, updating $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]}$ as $\mathcal{V}_{\mathrm{m}}^{[\mathrm{h}]} = \mathcal{V}_{\mathrm{m}}^{[\mathrm{h-1}]} \cup \mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]} \cup \mathcal{E}_{\mathrm{j}}^{[\mathrm{h}]}$ and discharging cluster $\mathcal{V}_{\mathrm{n}}^{[\mathrm{h-1}]}$. Please note that, during clustering (when cases A are more frequent than cases C), the number of overall clusters $\mathcal{V}$ can exceed the number of vine rows $\mathrm{n}_{\mathcal{V}}$ to be detected. This trend reverses in the last iterations, when the number of clusters $\mathcal{V}$ decreases as cases C are more frequent than cases A and B, finally settling to the detected vine rows number $\mathrm{n}_{\mathcal{V}}$. A graphical representation of the three iterations, examples of clustering Cases A, B and C, is reported in Fig. 2.4.

2.3.3 Key points refinement and sorting

With the final aim to define a continuous curve representing the spatial location of a single vine row, the key points of the cluster $\mathcal{V}_\mathrm{m}$ have to be refined and ordered from one end of the vine row to the other one. Indeed, in this phase, since a vine row can have any spatial layout in the map (e.g. curvilinear), a common interpolation procedure on an unordered set of points cannot be applied because the problem solution would not be an injective function.

A refinement is introduced in the last phase of the algorithm to decrease the number of key points $c \in \mathcal{V}_\mathrm{m}$ and by identifying the most representative ones (e_i), which will be then ordered, less dense and more equally spaced along the vine row. The proposed procedure is based on the following idea: one key point c_j is randomly selected from $\mathcal{V}_\mathrm{m}$ and then, starting from it, the vine row is scanned towards the two end points, defying the ordered set of enhanced key points e. More in detail, once the first enhanced key points $e_0 = c_\mathrm{j}$ was defined at iteration [0], the Local Reference Frame $\{\mathrm{Loc_j}\}$ is used to search for the closest point c_i along the positive axis $x^{\{\mathrm{Loc_j}\}}$, with a threshold $t = 2$ m between e_0 and c_i. The threshold t was introduced to avoid the selection of a point c too close to e_0. A new enhanced key point e_{+1} is thus defined at iteration [1] as $e_{+1} = c_\mathrm{i}$. The same procedure is performed along the negative direction of $x^{\{\mathrm{Loc_j}\}}$, finding the ~~enhanced~~ key point e_{-1}. An example of this task is reported in Fig. 2.5. The algorithm is performed until the two end points of the vine row $\mathcal{V}_{\mathrm{m}}$ are properly detected. Please not that, if at iteration $[\mathrm{k}]$ no points c are found as they overcome the threshold distance t, the threshold check is omitted and the most far point c from $e_{\mathrm{k}-1}$ is selected as end point on the vine row. When both the end points have been detected, the scanning procedure is then considered complete (Fig. 2.5c). The output of this processing phase for each vine row k is thus an ordered set of n_{k} enhanced key-points $E_{\mathrm{k}} = \{e_1, e_2, \ldots, e_{n_{\mathrm{k}}}\}_{\mathrm{k}}$, with e_1 and $e_{n_{\mathrm{k}}}$ being the two ends of the vine row as shown in Fig. 2.5d.

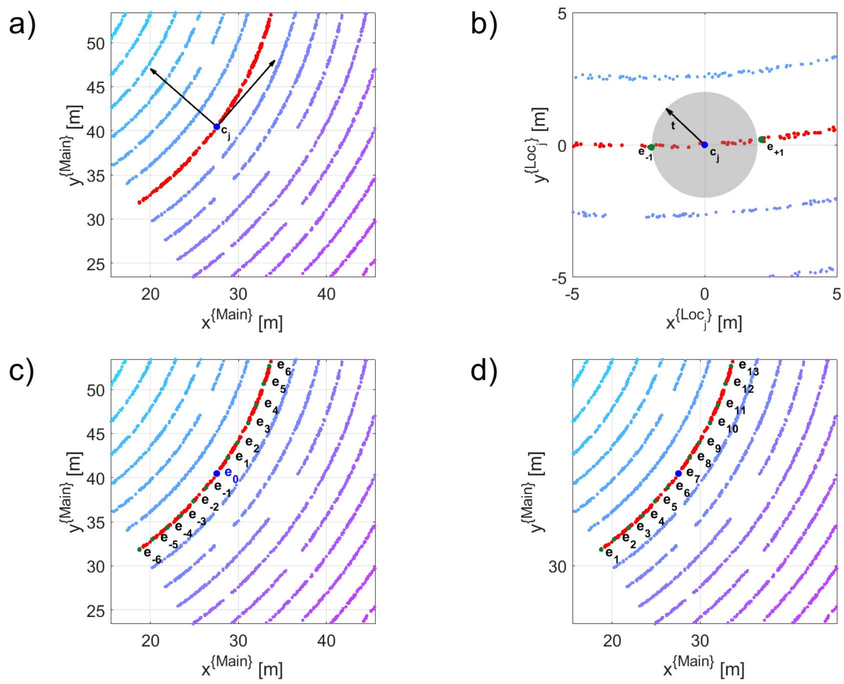


Figure 2.5. (a) clustered ($\mathcal{V}_{\mathrm{m}}$) comprehensive set of canopy centre points $\mathcal{C}$ (grey dots) and the Local Reference Frame $\{\mathrm{Loc_j}\}$ at the first enhanced key point $e_0 = c_{\mathrm{j}}$; (b) enlargement of subplot (a) at point c_{j} in the Local Reference Frame $\{\mathrm{Loc_j}\}$ having closest point e_{+1} along the

positive axis $x^{\{\mathrm{Loc_j}\}}$ and closest point e_{-1} along the negative axis $x^{\{\mathrm{Loc_j}\}}$ with a threshold $t = 2$ m; (c) scanning procedure completed as the two end points of the vine row $k = 7$ are properly detected and (d) final output of the processing phase for vine row $k = 7$ producing an ordered set of n_k enhanced key-points $E_\mathrm{k} = \{e_1, e_2, \dots, e_{n_\mathrm{k}}\}_\mathrm{k}$, with e_1 and e_{n_k} being the two vine row end-points.

2.3.4 Vine row localization

Considering each detected vine row, proposed algorithm provides an ordered set of enhanced key-points coordinates $E_\mathrm{k} = \{e_1, e_2, \dots, e_{n_\mathrm{k}}\}_\mathrm{k}$, with e_1 and e_{n_k} being the two vine row end-points, and a curve γ_k passing through them, representing the spatial layout of each vine row (Fig. 2.6). The spatial coordinates of points are represented both in local and WGS84 reference frames. In addition, the algorithm provides the vine row length, difference of altitude and average orientation, with respect to the west-east direction.

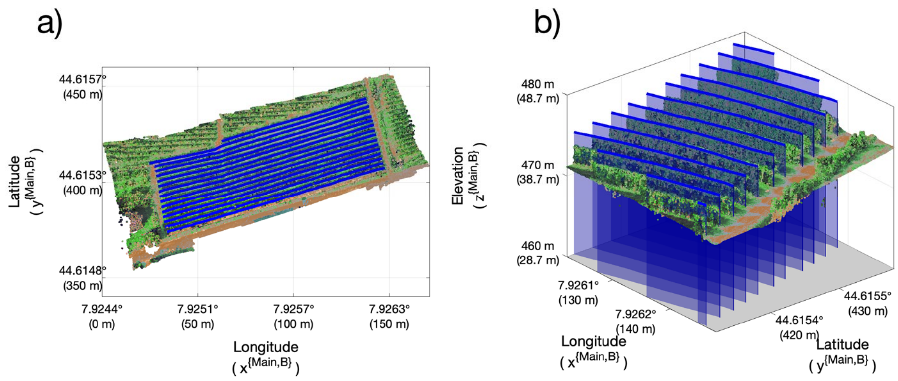


Figure 2.6. 2D (a) and 3D (b) graphical representation of the algorithm output obtained by processing parcel F (Fig. 2.1b). (in the 2D we use this image to plot the gamma lines (γ), with numbering and with one

row with point plotted and numbered) (in the 3D, we could plot the gamma lines over the plant (e.g. 10 meters high, with the vine row wall made by a patch)

2.4 Results and discussion

Detailed results on localisation information obtained by processing the point cloud of Parcel F are organized in Table 2.1, while the graphical representation of them is reported in Figure 2.6. Each detected vine row in a parcel is assigned an id to identify them and are ordered with respect to the west-east direction. Further outputs of the algorithm showcase individual vine row length and average orientation angle, difference of altitude, and the spatial coordinates of points representing the two end points (e_1 and e_{n_k}) of the vine row both in local and WGS84 reference frames. In addition, the table also provides the total number of enhanced key-points E_k that constitutes the curve γ_k, representing the spatial layout of each vine row.

Table 2.1. Algorithm output: location and information of each vine row of parcel F)

Vine row ID	Length [m]	Average orientation from E-W direction [deg]	Elevation difference [m]	End points locations [x, y, z] in $\{\mathbf{Main}, \mathbf{C}\}$ [m, m, m]	End points locations [lat, lon, alt] in $\{\mathbf{WGS84}\}$ [deg, deg, meter]	Total number of enhanced key-points, card$(\boldsymbol{E}_\mathrm{k})$

1 F	118.56	16	38.49	$e_1 = [33.97, 373.70, 36.42]^T$ $e_{58} = [147.55, 407.43, 40.83]^T$	$e_1 = [44.61503179, 7.924851157, 465.16]^T$ $e_{58} = [44.61533532, 7.926282013, 469.56]^T$	58
2 F	118.84	17	38.92	$e_1 = [32.85, 376.42, 36.66]^T$ $e_{60} = [146.77, 409.93, 41.21]^T$	$e_1 = [44.61505626, 7.924837047, 465.40]^T$ $e_{60} = [44.61535782, 7.926272187, 469.94]^T$	60
3 F	118.55	17	39.17	$e_1 = [32.28, 379.04, 36.71]^T$ $e_{58} = [145.98, 412.39, 41.48]^T$	$e_1 = [44.61507984, 7.924829865, 465.45]^T$ $e_{58} = [44.61537995, 7.926262234, 470.22]^T$	58
4 F	118.34	17	39.48	$e_1 = [31.79, 381.52, 36.86]^T$ $e_{60} = [145.28, 414.87, 41.75]^T$	$e_1 = [44.61510215, 7.924823691, 465.60]^T$ $e_{60} = [44.61540227, 7.926253415, 470.49]^T$	60
5 F	118.43	17	39.74	$e_1 = [31.27, 383.99, 37.10]^T$ $e_{61} = [144.87, 417.32, 42.00]^T$	$e_1 = [44.61512438, 7.924817140, 465.84]^T$ $e_{61} = [44.61542431, 7.926248249, 470.74]^T$	61
6 F	117.78	17	40.02	$e_1 = [31.11, 386.77, 37.41]^T$ $e_{60} = [144.15, 419.66, 42.32]^T$	$e_1 = [44.61514939, 7.924815123, 466.15]^T$ $e_{60} = [44.61544537, 7.926239179, 471.06]^T$	60
7 F	117.81	17	40.37	$e_1 = [30.30, 389.43, 37.79]^T$ $e_{60} = [143.48, 421.90, 42.58]^T$	$e_1 = [44.61517333, 7.924804918, 466.53]^T$ $e_{60} = [44.61546552, 7.926230738, 471.32]^T$	60
8 F	117.62	16	40.67	$e_1 = [30.02, 392.23, 38.23]^T$ $e_{60} = [142.98, 424.82, 42.96]^T$	$e_1 = [44.61519852, 7.924801390, 466.97]^T$ $e_{60} = [44.61549180, 7.926224438, 471.70]^T$	60
9 F	117.26	16	41.08	$e_1 = [29.53, 394.85, 38.67]^T$ $e_{60} = [142.98, 424.82, 42.96]^T$	$e_1 = [44.61522210, 7.924795216, 467.41]^T$ $e_{60} = [44.61551123, 7.926214864, 471.99]^T$	60
10 F	117.21	17	41.47	$e_1 = [29.25, 397.69, 39.08]^T$ $e_{59} = [141.85, 429.89, 43.65]^T$	$e_1 = [44.61524765, 7.924791687, 467.82]^T$ $e_{59} = [44.61553742, 7.926210202, 472.39]^T$	59
11 F	116.97	16	41.83	$e_1 = [28.73, 400.11, 39.34]^T$ $e_{59} = [141.16, 432.10, 43.95]^T$	$e_1 = [44.61526943, 7.924785136, 468.08]^T$ $e_{59} = [44.61555731, 7.926201509, 472.69]^T$	59

12 F	116.92	16	42.15	$e_1 = [28.04, 402.95, 39.70]^{\mathrm{T}}$ $e_{60} = [140.60, 434.30, 44.21]^{\mathrm{T}}$	$e_1 = [44.61529498, 7.924776442, 468.44]^{\mathrm{T}}$ $e_{60} = [44.61557710, 7.926194454, 472.95]^{\mathrm{T}}$	60
13 F	116.17	16	42.55	$e_1 = [27.85, 405.56, 40.10]^{\mathrm{T}}$ $e_{59} = [139.67, 436.75, 44.63]^{\mathrm{T}}$	$e_1 = [44.61531847, 7.924774048, 468.84]^{\mathrm{T}}$ $e_{59} = [44.61559915, 7.926182737, 473.37]^{\mathrm{T}}$	59
14 F	116.25	18	42.92	$e_1 = [27.46, 408.01, 40.44]^{\mathrm{T}}$ $e_{58} = [139.28, 439.36, 45.09]^{\mathrm{T}}$	$e_1 = [44.61534052, 7.924769134, 469.18]^{\mathrm{T}}$ $e_{58} = [44.61562263, 7.926177824, 473.83]^{\mathrm{T}}$	58
15 F	79.60	17	44.18	$e_1 = [62.08, 418.65, 41.32]^{\mathrm{T}}$ $e_{41} = [138.23, 441.72, 45.67]^{\mathrm{T}}$	$e_1 = [44.61543627, 7.925205269, 470.06]^{\mathrm{T}}$ $e_{41} = [44.61564387, 7.926164596, 474.411]^{\mathrm{T}}$	41
16 F	78.91	17	44.61	$e_1 = [62.21, 421.12, 41.49]^{\mathrm{T}}$ $e_{41} = [137.83, 443.59, 45.82]^{\mathrm{T}}$	$e_1 = [44.61545849, 7.925206906, 470.23]^{\mathrm{T}}$ $e_{41} = [44.61566069, 7.926159556, 474.56]^{\mathrm{T}}$	41

To assess the performance of the proposed vineyard 3D point cloud processing method, two families of quality indices were defined: the first one aimed at properly quantify the performance of vine rows detection (indices 1, 2 and 3) and the second to evaluate the accuracy of the vine rows position provided by the algorithm with respect to reference ones (indices 4, 5 and 6). Their definitions are reported in Table 2.2. As a reference, a set of line class objects were manually drawn using QGIS software, each one aligned with a single vine row. The procedure, which was performed on a plan view of the 3D point cloud, provided the latitude and longitude coordinate of each manually drawn point ($e_1^*, e_2^*, \dots, e_{\mathrm{m_k}}^*$) defining the line object. The altitude of each point was then retrieved by the local digital terrain model. Please note that the number of points representing the location of vine row k provided by the algorithm (n_{k}) and by the manual procedure ($\mathrm{m_k}$) may differ.

Table 2.2. Indices to evaluate the vine row detection and localisation results.

	Index name	Definition
Detection indices	1. Good detection	Percentage of properly detected vine rows with respect to the number of real vine rows
	2. Extra detection	Percentage of wrongly detected vine rows with respect to the number of real vine rows
	3. Missed detection	Percentage of not detected vine rows with respect to the number of real vine rows
Localisation indices	4. Euclidean distances between end points (DEP)	Average euclidean distances between each automatically detected end-point E_i and manual reference one E_i^*: $$DEP = \frac{1}{2}\sum_{i\in\{a,b\}} \lVert E_i - E_i^* \rVert_2$$
	5. Euclidean distances of enhanced key points (DEK)	Average distance of all automatically detected key points and the manual reference line: $$DEK = \frac{1}{n_\mathrm{k}}\sum_{i=1}^{N_j} \lVert e_i - p_i \rVert_2$$ where N_j is the number of points defining vine row j and p_i is the projection of e_i on the manual reference line (Fig. 2.7)

	6. Curves overlapping factor (COF)	Ratio of the area A_j of the region delimited by algorithms γ_j and manual lines γ_j^*, and the manually detected vine row length: $COF = \frac{A_j}{\sum_{i=1}^{N_j-1} \left\| e_{i+1}^* - e_i^* \right\|_2}$

Regarding the detection of vine rows, obtained results showed a *Good detection* index of 100%, and both *Extra detection* and *Missed detection* indices equal to 0%. This is related to the fact that all the 155 vine rows within the seven considered parcels were properly detected and no vine rows were wrongly found/located. The computation of these three detection quality indices (indices 1,2 and 3 of Table 2.2) were computed comparing the algorithm output with a visual inspection based on an in-field survey.

For what concern the accuracy of the vine row localization, the three indices *Euclidean Distances between End Points (DEP)* (index 4), *Euclidean Distances of Enhanced Key-points (DEK)* (index 5) and *Curves Overlapping Factor (COF)* (index 6) were computed for every detected vine row, and the average and standard deviation of the obtained values, grouped by processed parcels as well as the overall vine rows, are reported in Table 2.3. The average *DEP* index of each considered parcel varied between 0.07 and 0.17 meters, with standard deviations of 0.04 and 0.14, respectively. Considering the entire processed dataset, the average *DEP* index was 0.12 meters with a standard deviation of 0.10 meters. Obtained values of *DEP* index proved that the algorithm is able to properly detect the ends point of vine rows in an automatic way from a vineyard 3D point cloud. Considering the *DEK* index, obtained average values of each parcel were within the range 0.04 and 0.06 meters (with standard deviations of 0.01 and 0.01). The obtained error, which is really small, is compatible with the DGPS one. These prove that the accuracy of the detection of vine row end points provided by the algorithm is similar to ones obtained by the in field survey, which is the state of the art. The average *COF* index of the parcels varied between 0.03 and 0.06 meters with standard deviations of 0.01 and 0.01. Both the DEK and COF indices quantify the error between each vine row location, expressed by algorithm key points and manual ones. Obtained values proved that the accuracy of the algorithm is high in detecting the vine rows location along their whole extensions. All the quality indices values, which are in the order of few centimetres, show that the algorithm outputs, in term of vine row location, are compatible with requirement of precision agriculture operations, such as UGV path planning and autonomous guidance.

Table 2.3. Results of vine rows clustering and localisation error indices with respect to the manually detected reference vine rows applied on seven different parcels. Overall average error indices are reported in the last row.

	DEP [m]		**DEK [m]**		**COF [m]**	
	Average	**Std**	**Average**	**Std**	**Average**	**Std**
Parcel A	0.07	0.04	0.05	0.02	0.05	0.02
Parcel B	0.11	0.07	0.04	0.01	0.04	0.01
Parcel C	0.07	0.10	0.04	0.01	0.03	0.01
Parcel D	0.17	0.14	0.06	0.01	0.06	0.01
Parcel E	0.16	0.13	0.04	0.01	0.04	0.01
Parcel F	0.08	0.05	0.04	0.01	0.04	0.01
Parcel G	0.11	0.07	0.04	0.01	0.04	0.01
Overall	0.12	0.10	0.05	0.01	0.04	0.01

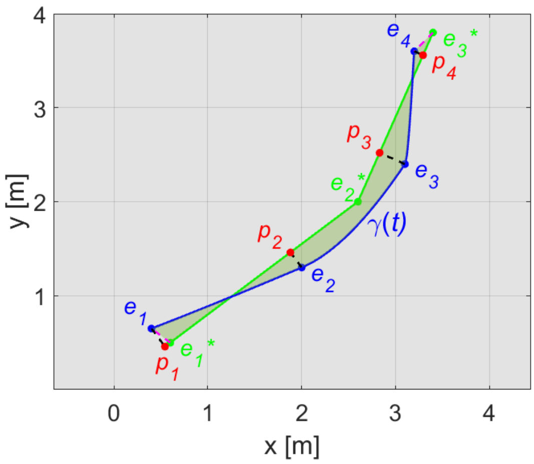


Figure 2.7. Sample algorithm output with ordered set of enhanced key-points $E_{\mathrm{k}} = \{e_1, e_2, \ldots, e_4\}_{\mathrm{k}}$ coordinates (blue points, with e_1 and e_4 being the vine row end-points) constituting a curve $\gamma(t)$ passing through them, representing the spatial layout of the vine row. The green set of points represent the manually generated reference key points whereas, the red points are the projection of reference key points on the curve $\gamma(t)$, used to evaluate Euclidean Distances of Enhanced Key-points. Whereas the green shaded area represents the region delimited by algorithms line and the manually detected vine row length line and is used to obtain the Curves Overlapping Factor.

2.5 Conclusions

In this chapter, an innovative/unsupervised algorithm that automatically clusters and localise individual vine rows within the 3D point clouds models of vineyard is presented. The proposed methodology clusters the 3D points intro groups representing individual vine rows and provides information about their spatial layout characterised by vine row end points and a curve following the centre of the row.

The robustness of the proposed algorithm was verified on 7 different vineyard parcels characterised by a sloped land formation with varying elevations. The proposed density-based clustering approach by means of an elliptic Region of Interest is not hindered by curvilinear vine rows or missing plants which are typical vineyard scenarios. The validation of the algorithm results was performed by comparing it to the manually detected vine rows using Matlab software (MathWorks ®, 2020).

The obtained results verified that the algorithm is able to cluster individual vine rows with 100 percent accuracy, while providing useful localisation information about the rows. This information is of crucial importance for infield autonomous machines 3D path planning to perform infield tasks with high accuracy, without damaging the crop. The possibility to automatically cluster and localise vine rows within a 3D point cloud map will lead the path to a new generation of unsupervised point-cloud processing algorithms aimed at evaluating crop status and developing new procedures for precision agriculture applications.

2.6 References

- Bechar, A., Vigneault, C., 2017. Agricultural robots for field operations. Part 2: Operations and systems. Biosystems Engineering, 153, 110-128. https://doi.org/10.1016/j.biosystemseng.2016.11.004
- Blanquart, J.-E., Sirignano, E., Lenaerts, B., Saeys, W., 2020. Online crop height and density estimation in grain fields using LiDAR. Biosystems Engineering, 198, 1-14. https://doi.org/10.1016/j.biosystemseng.2020.06.014

- Campos, J., Llop, J., Gallart, M., García-Ruiz, F., Gras, A., Salcedo, R., Gil, E., 2019. Development of canopy vigour maps using UAV for site-specific management during vineyard spraying process. Precis. Agric. 20, 1136-1156, https://doi.org/10.1007/s11119-019-09643-z
- Chakraborty, M., Khot, L.R., Sankaran, S., Jacoby, P.W., 2019. Evaluation of mobile 3D light detection and ranging based canopy mapping system for tree fruit crops. Computers and Electronics in Agriculture, 158 284-293. https://doi.org/10.1016/j.compag.2019.02.012
- Chen, Y., Zhang, B., Zhou, J., Wang, K., 2020. Real-time 3D unstructured environment reconstruction utilizing VR and Kinect-based immersive teleoperation for agricultural field robots. Computers and Electronics in Agriculture, 175, 105579. https://doi.org/10.1016/j.compag.2020.105579
- Colorado, J.D., Cera-Bornacelli, N., Caldas, J.S., Petro, E., Rebolledo, M.C., Cuellar, D., et al., 2020. Estimation of nitrogen in rice crops from UAV-captured images. Remote Sensing, 12, 3396, https://doi.org/10.3390/rs12203396
- Comba, L., Biglia, A., Ricauda Aimonino, D., Gay, P., 2018. Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. Comput. Electron. Agr. 155, 84-95, https://doi.org/10.1016/j.compag.2018.10.005 nel paragrafo 2
- Comba, L., Biglia, A., Ricauda Aimonino, D., Barge, P., Tortia, C., Gay, P., 2019. 2D and 3D data fusion for crop monitoring in precision agriculture. IEEE International Workshop on Metrology for Agriculture and Forestry, 62-67, https://doi.org/10.1109/MetroAgriFor.2019.8909219

- Comba, L., Biglia, A., Ricauda Aimonino, D., Tortia, C., Mania, E., Guidoni, S., et al., 2020a. Leaf Area Index evaluation in vineyards using 3D point clouds from UAV imagery. Precis. Agric. 21, 881-896, https://doi.org/10.1007/s11119-019-09699-x
- Comba, L., Zaman, S., Biglia, A., Ricauda Aimonino, D., Dabbene, F., Gay, P., 2020b. Semantic interpretation and complexity reduction of 3D point clouds of vineyards. Biosyst. Eng. 197, 216-230, https://doi.org/10.1016/j.biosystemseng.2020.05.013
- Condotta, I.C.F.S., Brown-Brandl, T.M., Pitla, S.K., Stinn, J.P., Silva-Miranda, K.O., 2020. Evaluation of low-cost depth cameras for agricultural applications. Comput. Electron. Agr. 173, 105394, https://doi.org/10.1016/j.compag.2020.105394
- Cuevas-Velasquez, H., Gallego, A.-J., Fisher, R.B., 2020. Segmentation and 3D reconstruction of rose plants from stereoscopic images. Comput. Electron. Agr. 171, 105296, https://doi.org/10.1016/j.compag.2020.105296
- Dusadeerungsikul, P.O., Nof, S.Y., 2019. A collaborative control protocol for agricultural robot routing with online adaptation. Computers & Industrial Engineering, 135, 456-466. https://doi.org/10.1016/j.cie.2019.06.037
- Escolà, A., Martínez-Casasnovas, J.A., Rufat, J., Arnó, J., Arbonés, A., Sebé, F., Pascual, M., Gregorio, E., Rosell-Polo, J.R., 2017. Mobile terrestrial laser scanner applications in precision fruticulture/horticulture and tools to extract information from canopy point clouds. Precision Agriculture, 18 (1), 111-132. https://doi.org/10.1007/s11119-016-9474-5

- Ester, M., Kriegel, H.-P., Sander, J., Xiaowei, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining, 226-231
- Feng, A., Zhou, J., Vories, E.D., Sudduth, K.A., Zhang, M., 2020. Yield estimation in cotton using UAV-based multi-sensor imagery. Biosyst. Eng. 193, 101-114, https://doi.org/10.1016/j.biosystemseng.2020.02.014
- Gao, G., Xiao, K., Jia, Y., 2020. A spraying path planning algorithm based on colour-depth fusion segmentation in peach orchards. Comput. Electron. Agr. 173, https://doi.org/105412, 10.1016/j.compag.2020.105412
- Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Morros, J.-R., Ruiz-Hidalgo, J., Vilaplan, V., et al., 2020. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agr. 169, 105165, https://doi.org/10.1016/j.compag.2019.105165
- Gil, E., Llorens, J., Llop, J., Fàbregas, X., Escolà, A., Rosell-Polo, J.R., 2013. Variable rate sprayer. Part 2 – Vineyard prototype: Design, implementation, and validation. Computers and Electronics in Agriculture, 95, 136-150. https://doi.org/10.1016/j.compag.2013.02.010
- Graf Plessen, M.M., Bemporad, A., 2017. Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines. Biosystems Engineering, 153, 28-41. https://doi.org/10.1016/j.biosystemseng.2016.10.019
- Hameed, I.A., la Cour-Harbo, A., Osen, O.L., 2016. Side-to-side 3D coverage path planning approach for agricultural robots to minimize

skip/overlap areas between swaths. Robot. Auton. Syst. 76, 36-45, https://doi.org/10.1016/j.robot.2015.11.009

- Jin, J., Tang, L., 2011. Coverage path planning on three-dimensional terrain for arable farming. J Field Robot. 28, 424-440, https://doi.org/10.1002/rob.20388
- Jurado, J.M., Pádua, L., Feito, F.R., Sousa, J.J., 2020. Automatic grapevine trunk detection on UAV-based point cloud. Remote Sens. 12, 3043, https://doi.org/10.3390/rs12183043
- Kang, H., Chen, C., 2020. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. Comput. Electron. Agr. 171, 105302, https://doi.org/10.1016/j.compag.2020.105302
- Kerkech, M., Hafiane, A., Canals, R., 2020. Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach. Comput. Electron. Agr. 174, 105446, https://doi.org/10.1016/j.compag.2020.105446
- Khajepour, A., Sheikhmohammady, M., Nikbakhsh, E., 2020. Field path planning using capacitated arc routing problem. Computers and Electronics in Agriculture, 173, 105401. https://doi.org/10.1016/j.compag.2020.105401
- Khaliq, A., Comba, L., Biglia, A., Ricauda Aimonino, D., Chiaberge, M., Gay, P., 2019. Comparison of satellite and UAV-based multispectral imagery for vineyard variability assessment. Remote Sens. 11, 436, https://doi.org/10.3390/rs11040436
- Li, Y., Iida, M., Suyama, T., Suguri, M., Masuda, R., 2020. Implementation of deep-learning algorithm for obstacle detection and collision avoidance

for robotic harvester. Computers and Electronics in Agriculture, 174, 105499. https://doi.org/10.1016/j.compag.2020.105499

- Llorens, J., Gil, E., Llop, J., Escolà, A., 2011. Ultrasonic and LiDAR sensors for electronic canopy characterization in vineyards: Advances to improve pesticide application methods. Sensors, 11 (2), 2177-2194, https://doi.org/10.3390/s110202177
- Matese, A., Cinat, P., Romboli, Y., Berton, A., Di Gennaro, S.F., 2019. Missing plant detection and biomass estimation from 3D models generated from UAV in a vineyard. Precision agriculture ’19, 165 – 172. https://doi.org/10.3920/978-90-8686-888-9
- MathWorks Matlab®, 2020b from https://www.mathworks.com/help/stats/dbscan-clustering.html
- Malavazi, F.B.P., Guyonneau, R., Fasquel, J.-B., Lagrange, S., Mercier, F., 2018. LiDAR-only based navigation algorithm for an autonomous agricultural robot. Computers and Electronics in Agriculture,154, 71-79. https://doi.org/10.1016/j.compag.2018.08.034
- Mazzia, V., Comba, L., Khaliq, A., Chiaberge, M., Gay, P., 2020. UAV and Machine Learning Based Refinement of a Satellite-Driven Vegetation Index for Precision Agriculture. Sensors, 20 (9), 2530-2546. https://doi.org/10.3390/s20092530
- McAllister, W., Osipychev, D., Davisc, A., Chowdhary, G., 2019. Agbots: Weeding a field with a team of autonomous robots. Computers and Electronics in Agriculture, 163, 104827. https://doi.org/10.1016/j.compag.2019.05.036
- Michels, M., von Hobe, C.-F., Musshoff, O., 2020. A trans-theoretical model for the adoption of drones by large-scale German farmers. Journal

of Rural Studies, 75, 80-88. https://doi.org/10.1016/j.jrurstud.2020.01.005

- Mortensen, A.K., Bender, A., Whelan, B., Barbour, M.M., Sukkarieh, S., Karstoft, H., et al., 2018. Segmentation of lettuce in coloured 3D point clouds for fresh weight estimation. Comput. Electron. Agr. 154, 373-381, https://doi.org/10.1016/j.compag.2018.09.010
- Nagasaka, Y., Saito, H., Tamaki, K., Seki, M., Kobayashi, K., Taniwaki, K., 2009. An autonomous rice transplanter guided by global positioning system and inertial measurement unit. Journal of Field Robotics, 26 (6–7), 537-548. https://doi.org/10.1002/rob.20294
- Peng C., Vougioukas, S.G., 2020. Deterministic predictive dynamic scheduling for crop-transport co-robots acting as harvesting aids. Computers and Electronics in Agriculture, 178, 105702. https://doi.org/10.1016/j.compag.2020.105702
- Primicerio, J., Caruso, G., Comba, L., Crisci, A., Gay, P., Guidoni, S., Genesio, L., Ricauda Aimonino, D., Primo Vaccari, F., (2017). Individual plant definition and missing plant characterization in vineyards from high-resolution UAV imagery. European Journal of Remote sensing, 50 (1), 179-186. https://doi.org/doi:10.1080/22797254.2017.1308234
- Rao Mogili, U.M., Deepak, B.B.V.L., 2018. Review on Application of Drone Systems in Precision Agriculture. Procedia Computer Science, 133, 502-509. https://doi.org/10.1016/j.procs.2018.07.063
- Sanz, R., Llorens, J., Escolà, A., Arnó, J., Planas, S., Román, C., Rosell-Polo, J.R., 2018. LIDAR and non-LIDAR-based canopy parameters to estimate the leaf area in fruit trees and vineyard. Agricultural and Forest

Meteorology, 260–261, 229-239. https://doi.org/10.1016/j.agrformet.2018.06.017

- Shendryk, Y., Sofonia, J., Garrard, R., Rist, Y., Skocaj, D., Thorburn, P., 2020. Fine-scale prediction of biomass and leaf nitrogen content in sugarcane using UAV LiDAR and multispectral imaging. Int. J. Appl. Earth Obs. 92, 102177, https://doi.org/10.1016/j.jag.2020.102177
- Sun et al., 2020. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. ISPRS Journal of Photogrammetry and Remote Sensing, 160, 195–207. https://doi.org/10.1016/j.isprsjprs.2019.12.011
- Thompson, L.J., Puntel, L.A., 2020. Transforming unmanned aerial vehicle (UAV) and multispectral sensor into a practical decision support system for precision nitrogen management in corn. Remote Sensing, 12(10), 1597; https://doi.org/10.3390/rs12101597
- Wanga, D., Li, W., Liu, X., Li, N., Zhang, C., 2020. UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. Computers and Electronics in Agriculture, 175, 105523. https://doi.org/10.1016/j.compag.2020.105523
- Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J Photogramm 105, 286-304, https://doi.org/10.1016/j.isprsjprs.2015.01.016
- Wu, G., Li, B., Zhu, Q., Huang, M., Guo, Y., 2020. Using color and 3D geometry features to segment fruit point cloud and improve fruit

recognition accuracy. Comput. Electron. Agr. 174, 105475, https://doi.org/10.1016/j.compag.2020.105475

- Xie, C., Yang, C., 2020. A review on plant high-throughput phenotyping traits using UAV-based sensors. Comput. Electron. Agr. 178, 105731, https://doi.org/10.1016/j.compag.2020.105731
- Zahid, A., Mahmud, Md S., He, L., Choi, D., Heinemann, P., Schupp, J., 2020. Development of an integrated 3R end-effector with a cartesian manipulator for pruning apple trees. Computers and Electronics in Agriculture, 179, 105837. https://doi.org/10.1016/j.compag.2020.105837
- Zeng et al., 2020. Semantic segmentation of sparse 3D point cloud based on geometrical features for trellis-structured apple orchard. Biosystems engineering, 196, 46-55. https://doi.org/10.1016/j.biosystemseng.2020.05.015
- Zhang, B., Y., Xie, J., Zhou, K. ,Wang, Z., Zhang, 2020a. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. Computers and Electronics in Agriculture, 177, 105694. https://doi.org/10.1016/j.compag.2020.105694
- Zhang, C., Zhang, K., Ge, L., Zou, K., Wang, S., Zhang, J., Li, W., 2020b. A method for organs classification and fruit counting on pomegranate trees based on multi-features fusion and support vector machine by 3D point cloud. Scientia Horticulturae, XX, 109791. https://doi.org/10.1016/j.scienta.2020.109791

- Zhang, C., Kovacs, J., 2012. The application of small unmanned aerial systems for precision agriculture: a review. Precision Agriculture 13, 693–712. https://doi.org/10.1007/s11119-012-9274-5
- Zhou, J., Fu, X., Zhou, S., Zhou, J., Ye, H., Nguyen, H.T., 2019. Automated segmentation of soybean plants from 3D point cloud using machine learning. Computers and Electronics in Agriculture, 162, 143-153. https://doi.org/10.1016/j.compag.2019.04.014

3. Semantic interpretation and complexity reduction of 3D point clouds of vineyards

Abstract

In precision agriculture, autonomous ground and aerial vehicles can lead to favourable improvements in field operations, extending crop scouting to large fields and performing field tasks in a timely and effective way. However, automated navigation and operations within a complex scenario require specific and robust path planning and navigation control. Thus, in addition to proper knowledge of their instantaneous position, robotic vehicles and machines require an accurate spatial description of their environment. In this chapter an innovative modelling framework is presented to semantically interpret 3D point clouds of vineyards and to generate low complexity 3D mesh models of vine rows. The proposed methodology, based on a combination of convex hull filtration and minimum area c-gon design, reduces the number of instances required to describe the spatial layout and shape of vine canopies allowing the amount of data to be reduced without losing relevant crop shape information. The algorithm is not hindered by complex scenarios, such as non-linear vine rows, as it is able to automatically process non uniform vineyards. Results demonstrated a data reduction of about 98%; from the 500 Mb ha-1 required to store the original dataset to 7.6 Mb ha-1 for the low complexity 3D mesh. Reducing the amount of data is crucial to reducing computational times for large original datasets, thus enabling the exploitation of 3D point cloud information in real-time during field operations. When considering scenarios involving cooperating machines

and robots, data reduction will allow rapid communication and data exchange between in field actors.

Keywords: Precision agriculture; Photogrammetry; Big data; UAV remote sensing; Semantic interpretation; 3D point cloud segmentation.

Nomenclature

a	Dimensions of vine row section $\mathcal{S}_\mathrm{k}$ along x_k axis [m] (model parameter)
b	Distance between two sequential vine row sections $\mathcal{S}_\mathrm{k}$ and $\mathcal{S}_{\mathrm{k+1}}$[m] (model parameter)
c	Number of vertices of c-gon $\mathcal{P}^c$ (model parameter)
R_k	*Complexity reduction* index of 3D mesh $\mathcal{M}$
$\mathcal{C}_\mathrm{k}$	Set of points representing the k$^{\text{th}}$ canopy section ($\mathcal{C}_k = \mathcal{C}_k^- \cup \mathcal{C}_k^+$)
$\mathcal{C}^{2\mathrm{D}}$	Two-dimensional projection of $\mathcal{C}$ on the plane $x = 0$
$\tilde{\mathcal{C}}^{2\mathrm{D}}$	Outlier-filtered $\mathcal{C}^{2\mathrm{D}}$ set of points
$F_{\mathrm{k,k+1}}$	Set of triangular faces of the generated model mesh, between vertices V_k and $V_{\mathrm{k+1}}$
G_k	*Good-modelling* index of 3D mesh $\mathcal{M}$
h_k	Peak location of H_z
$H_\mathrm{y}(\mathcal{S}_\mathrm{k}, s)$	Normalised frequencies distribution histogram of points $p_\mathrm{i} \in \mathcal{S}_\mathrm{k}$ along y_k axes
$H_\mathrm{z}(\mathcal{S}_\mathrm{k}, t)$	Normalised frequencies distribution histogram of points $p_\mathrm{i} \in \mathcal{S}_\mathrm{k}$ along z_k axes
$\mathcal{H}$	Convex hull of points set $\mathcal{C}^{2\mathrm{D}}$
$\mathcal{K}$	Set of all the considered vine row section $\mathcal{S}_\mathrm{k}$
L'_k and L''_k	Lines defining plane $\wp_\mathrm{k}$
$\mathcal{M}$	Low complexity 3D triangulated mesh of vine rows
$N_\mathcal{C}$	Set of points $\mathcal{C}$ cardinality
$N_\mathcal{H}$	Cardinality of vertices U^{2D} of the convex hull $\mathcal{H}$
$N_{\mathcal{PC}}$	Cardinality of point-cloud $\mathcal{PC}$
O_k	*Over-modelling* index of 3D mesh $\mathcal{M}$

$\boldsymbol{O}_{\mathrm{LOC_k}}^{\{\mathrm{WGS84}\}}$	Origin of local reference frame $\mathrm{LOC_k}$ in $\mathrm{WGS84}$ coordinates
$\mathcal{P}^c$	c-gon containing the point set $\tilde{\mathcal{C}}^{2\mathrm{D}}$ with vertices
$\mathcal{P}[V^{c*}]$	Minimum area c-gon containing the point set $\tilde{\mathcal{C}}^{2\mathrm{D}}$
$\mathcal{PC}$	3D point cloud of vineyard
$\wp_{\mathrm{k}}^{+}$	Plane defined by two lines L_{k}' and L_{k}''
Q_{k}	Model $\mathcal{M}_{\mathrm{k}}$ quality score
s_y	Bin of the histogram H_{y}
s_z	Bin of the histogram H_{z}
$\mathcal{S}_{\mathrm{k}}$	Subset of points representing a section of vine row
$\mathcal{S}_{\mathrm{k}}^{+}$ and $\mathcal{S}_{\mathrm{k}}^{-}$	Two sides of vine row section $\mathcal{S}_{\mathrm{k}}$ with $y \geq 0$ and $y < 0$ respectively
$\boldsymbol{u}_{\mathrm{i}}$	i-th vertex of convex hull $\mathcal{H}$
U_{k}	*under-modelling* index of 3D mesh $\mathcal{M}$
U^{2D}	set of vertices of the convex hull $\mathcal{H}$ in the 2D plane x=0
$\boldsymbol{v}_{\mathrm{i}}$	i-th vertex of c-gon $\mathcal{P}^c$ in the 2D plane x=0
$\mathfrak{v}_{\mathrm{i}}$	i-th vertex of c-gon $\mathcal{P}^c$ in the 3D space and of the 3D mesh $\mathcal{M}$
V^{2D}	Set of vertices of c-gon (polygon) $\mathcal{P}^c$ in the 2D plane x=0
V_{ref}	$\mathcal{C}_{\mathrm{k}}$ envelope volume
V_k	Set of vertices of c-gon (polygon) $\mathcal{P}^c$ in the 3D space
w_{k}	Peak location of H_{y}
x_{k}	x axis of the $\{\mathrm{LOC}\}$, tangent to the local wine row direction ϑ_k
y_{k}	y axis of the $\{\mathrm{LOC}\}$
$y_{\mathrm{k,max}}$	Greatest value of y coordinates of points in $\mathcal{S}_{\mathrm{k}}$
Y_{k}	Bins set of the histogram H_{y}
z_{k}	z vertical axis of the $\{\mathrm{LOC}\}$
$z_{\mathrm{k,max}}$	Greatest value of z coordinates of points in $\mathcal{S}_{\mathrm{k}}$
Z_{k}	Bins set of the histogram H_{z}

Greek letters

φ_{i}	Latitude coordinates of the i^{th} point of the 3D point cloud [°]
λ_{i}	Longitude coordinates of the i^{th} point of the 3D point cloud [°]
e_{i}	Elevation coordinates of the i^{th} point of the 3D point cloud [°]
ϑ_{k}	Local vine row orientation [°]
δ_{k}	Local inter row spacing along y_{k} axis [m]
δ_{s}	Bin width of histograms H_{y} and H_{z}
γ	Vine row centre line

Acronyms

2D	Two-dimensional
3D	Three-dimensional
GIS	Geographic information systems
UAV	Unmanned aerial vehicle
SfM	Structure from Motion
VSP	Vertical Shoot Position
$\{\mathrm{LOC}\}$	Local metrical reference frame
{WGS84}	World geodetic system 1984

3.1 Introduction

Precision agriculture has proven to be effective in increasing field productivity and product quality by optimising the efficiency of agricultural and management operations (Gebbers & Adamchuk, 2010; Tenhunen et al., 2019). This is achieved by the timely monitoring of crops and by performing site-specific operations (Reza et al., 2019; Sozzi et al., 2019; Khaliq et al, 2019; Comba et al., 2019a), whilst minimising the use of resources (Higgins et al., 2019; Peng et al., 2019) and improving environmental protection (Oberti et al., 2016; Grella et al., 2017). In this context, autonomous ground and aerial vehicles can lead to favourable improvements to precision agriculture operations, allowing crop scouting to be extended to large fields or uneven terrains and to improve management by timely performing in field tasks (Primicerio et al., 2017; Grimstad & From, 2017; Utstumo et al., 2018; Comba et al., 2019b), including with collaborative architectures (Campos et al., 2019). Moreover, in order to be competitive, robotic technology for agriculture should be reliable and cost-effective (Comba et al., 2016; Reina et al., 2017; Zaman et al., 2019).

However, partially/fully autonomous navigation and operations within a complex, irregular and unstructured scenarios, require developing specific algorithms for effective path planning and navigation, and to act on crops (Vidoni et al., 2016). To do this, in addition to proper knowledge of their instantaneous spatial position, robotic vehicles and machines require an accurate spatial description of the environment in which they are operating, e.g. inter-row width and crop canopy position and shape to avoid damage (Kassler, 2001; Van et al., 2013; Primicerio et al., 2015; Wang et al., 2019) and to profitably complete the tasks (Bechar & Vigneault, 2017).

Recently, enhanced performances have been achieved by three dimensional path planning which resulted in, for example, collision free paths from 3D obstacles (Han, 2018) and defined new strategies for field coverage, which overcomes the problems of standard 2D coverage (Hameed et al., 2016). This requires the development of new 3D models, such as point clouds or triangulated meshes (Weiss & Biber, 2011; Miranda-Fuentes et al., 2015). A raw 3D point cloud is a set of points, in an arbitrary 3D coordinate system, representing the visible surfaces of objects.

A 3D point cloud can be generated using 3D sensors or by photogrammetry using structure from motion (SfM) software, processing appropriate sets of 2D images. In agricultural applications, several studies have derived 3D crop models using 3D sensors, such as the light detection and ranging systems (LiDAR) (Mack et al., 2017) and by a family of devices known as depth cameras (Condotta et al., 2020). Depth cameras applied in agriculture can be based on three different technologies: stereoscopy (Luo et al., 2016), structured light (Saberioon & Cisar, 2016), and time-of-flight (Rosell-Polo et al., 2017; Bao et al., 2019). To derive 3D point clouds using SfM algorithms, several approaches have been investigated; exploiting images acquired by several cameras and involving RGB, multispectral, hyperspectral or thermal sensors (Feng et al., 2020). The significant developments in UAVs and remote sensors has increased the potential, and reduced the costs, of acquiring aerial imagery and, thus the generation of high density 3D point clouds of crops (Maes & Steppe, 2019; Wijesingha et al., 2019). In agriculture, this new modelling representation can facilitate comprehension of the environment, but proper algorithms for detecting and mapping crops and identifying soil and obstacles are needed (Mortensen et al., 2018; Comba et al., 2018). This task is not trivial since large 3D models of crops, including remotely sensed imagery and measurements made using in-field or on-vehicle sensors, require new processing algorithms to process big data (Wolfert et al., 2017; Van Evert et al., 2017; Pavón-Pulido et al., 2017; Zeybek & Şanlıoğlu, 2019). Also, these huge data sets contain a lot of information that requires appropriate data extraction approaches, depending on the required final goal (Serazetdinova et al., 2019).

In this chapter is presented an innovative modelling framework to semantically interpret 3D point clouds of vineyards and to generate low complexity 3D mesh models of vine rows. The proposed methodology reduces the amount of instances required to properly describe the spatial layout and shape of vine canopies; this allows the amount of data to be drastically reduced without losing relevant crop shape information. This is a crucial task that allows shorter computational times for the processing of large datasets (e.g. raw 3D point clouds representing crops), thereby enabling the exploitation of point clouds information in real time in the field. When considering cooperating machines and scenarios including robots, data reduction is relevant for enabling rapid communication and data exchange between in field actors. Moreover, the proposed modelling framework is not hindered by complex scenarios, such as hilly regions and/or non-linear vine rows, to enable it to automatically process information from non-uniform vineyards.

This chapter is structured as follows: section 2.2 presents the proposed modelling framework to generate vine rows using low complexity 3D meshes. The results in terms of modelling performance and quality, were evaluated on more than 128 m of vine rows, are presented in section 2.3, while section 2.4 reports the conclusions and future developments.

3.2 Materials and methods

The method used to reduce the complexity of the 3D point clouds can be divided into three main processing steps: (1) the extraction of a 3D point cloud subset representing a vineyard section, (2) the classification of the subset points into canopy and inter-row terrain categories (semantic interpretation) and, finally, (3) the canopy model simplification by determining an optimal polygon and generating a low complexity 3D mesh of the canopy (Fig. 3.1).

As previously discussed, the proposed methodology starts from a raw 3D point cloud, which is given by a set of $N_{\mathcal{PC}}$ points, representing the external surface of the objects, defined as

$$\mathcal{PC}^{\{\mathrm{WGS84}\}} = \{[\varphi_{\mathrm{i}}, \lambda_{\mathrm{i}}, e_{\mathrm{i}}]^{\mathrm{T}} \in \mathbb{R}^3;\ \mathrm{i} = 1, \dots, N_{\mathcal{PC}}\}, \tag{1}$$

where φ_{i}, λ_{i} and e_{i} are, respectively, the latitude, longitude and elevation coordinates of the i^{th} point of the 3D point cloud, measured in the World Geodetic System 1984 {WGS84}. The point cloud was obtained by processing UAV-based aerial images using a SfM algorithm (Agisoft Photoscan®, 2018, St. Petersburg, Russia),. In particular, a Parrot Sequoia® multispectral camera (Parrot©, 2018, Paris, France) was used to acquire more than 1,000 aerial images with a resolution of 1280 × 960 pixels. The UAV flight took place in Serralunga d'Alba (Piedmont, North-west Italy) on a vineyard of about 2.5 ha with latitude and longitude positions ranging between [44.62334 44.62539] and [7.99855 8.00250]. The vineyard was located on sloped land with an elevation ranging from 330 m to 420 m above sea level and a predominantly southwest orientation. Parcels were cultivated with Cv. Nebbiolo grapevine using a Vertical Shoot Position (VSP) trellis systems, with wine spacing of 0.9 m and inter row space of about 2.5 m. The height of the UAV flight was maintained close to 35 m with respect to the terrain by using a set of waypoints, which were defined on the basis of the vineyard Geographic Information System (GIS) map. A forward and side overlap greater than 80% was guaranteed between adjacent images. Prior to the images block alignment, a radiometric calibration was performed on the images by using the reference images of a Micasense calibrated reflectance panel (Seattle, Washington, USA) acquired before and after the UAV flight.

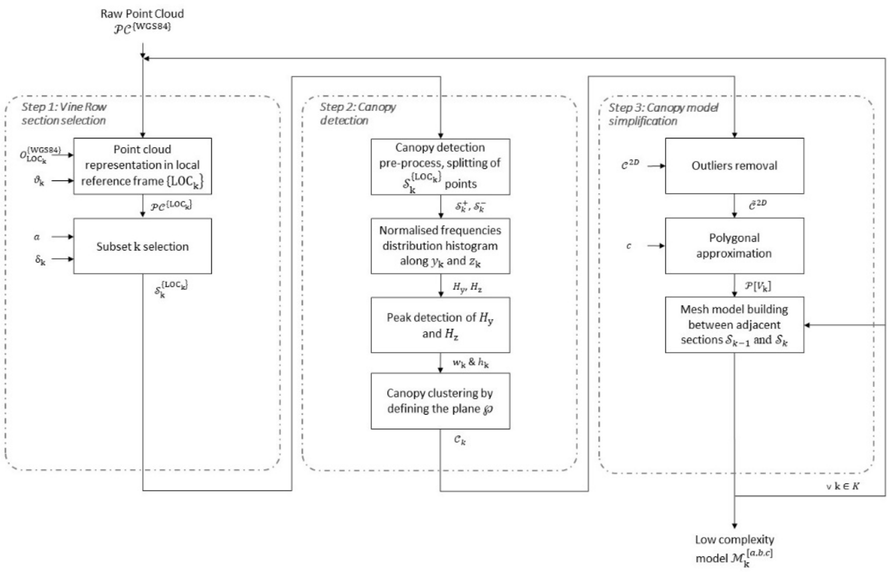


Fig. 3.1. Scheme of the defined modelling framework to generate low complexity 3D mesh models of vine rows from raw 3D point clouds.

3.2.1 Vine row section from raw 3D point cloud

In order to allow the proposed modelling framework to process the vineyards 3D point cloud with a broad set of characteristics, such as rectilinear and/or curvilinear layouts, or vineyards grown on flat and/or sloped terrain, the first processing step consists in properly selecting a subset $\mathcal{S}_{\mathrm{k}}$ representing a vine row section from the whole $\mathcal{PC}^{\{\mathrm{WGS84}\}}$ (Fig. 3.2).

This process was performed by defining a local metrical reference frame $\{\mathrm{LOC_k}\}$ by using the information on the vine row position, as provided by local vine row orientation ϑ_{k} and local inter row spacing δ_{k}, which are automatically provided by algorithms presented in Comba et al. (2018). The vine row position was defined as the parametrised curve $\gamma: t \in [0,1] \rightarrow \mathbb{R}^3$, which represents the canopy centre curve at soil level (Fig. 3.3). The origin of $\{\mathrm{LOC_k}\}$ was defined in $\boldsymbol{O}_{\mathrm{LOC_k}}^{\{\mathrm{WGS84}\}} \in \gamma$, so that the distance along the vine row centre line γ between two local reference systems $\{\mathrm{LOC_{k-1}}\}$ and $\{\mathrm{LOC_k}\}$, and thus between two vineyard subsets $\mathcal{S}_{\mathrm{k-1}}$ and $\mathcal{S}_{\mathrm{k}}$, is equal to b, satisfying the line integral

$$\int_{t_{\mathrm{k-1}}}^{t_{\mathrm{k}}} \|\gamma'(t)\| \mathrm{d}t = b \qquad (2)$$

where $\gamma(t_{\mathrm{k-1}}) = \boldsymbol{O}_{\mathrm{LOC_{k-1}}}$ and $\gamma(t_{\mathrm{k}}) = \boldsymbol{O}_{\mathrm{LOC_k}}$. The x_{k} axis of $\{\mathrm{LOC_k}\}$ was defined as tangent to line γ (local wine row direction ϑ_{k}), the z_{k} axis was defined as vertical and, finally, the y_{k} axis completes the Cartesian reference system (Fig. 3.3).

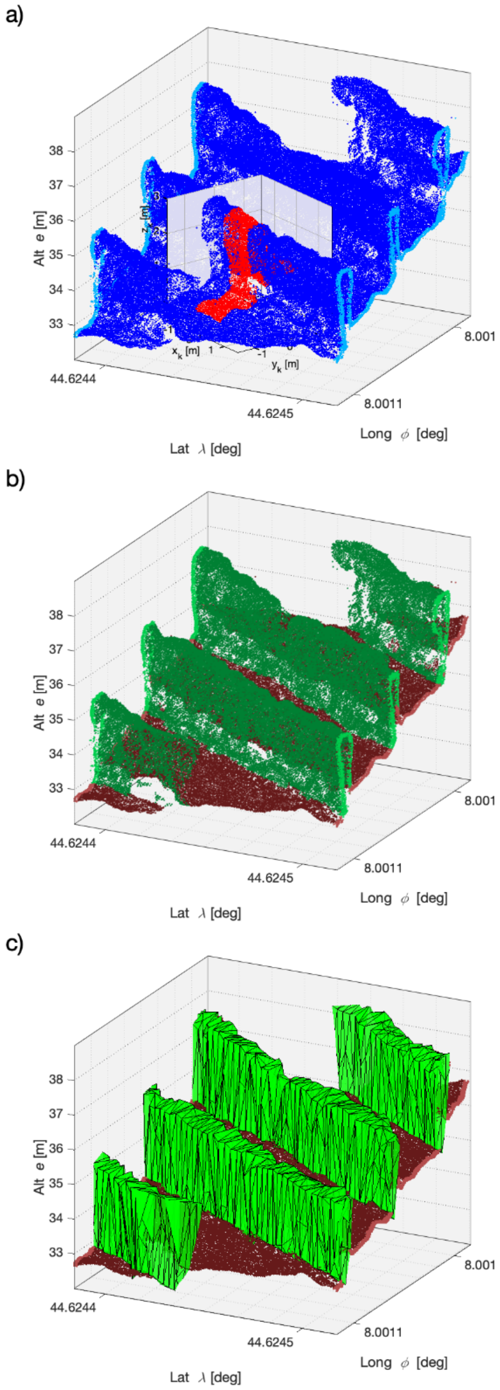


Fig. 3.2. (a) Portion of the raw 3D point cloud $\mathcal{PC}^{\{\mathrm{WGS84}\}}$ (blue) and sample vine row section $\mathcal{S}_{268}$ ($a = 0.8\ m$) (red); (b) canopy points $\mathcal{C}_k$ clustered (green) from the ones representing the inter row terrain

(brown); and (c) low complexity triangulated 3D mesh model $\mathcal{M}$ (light green).

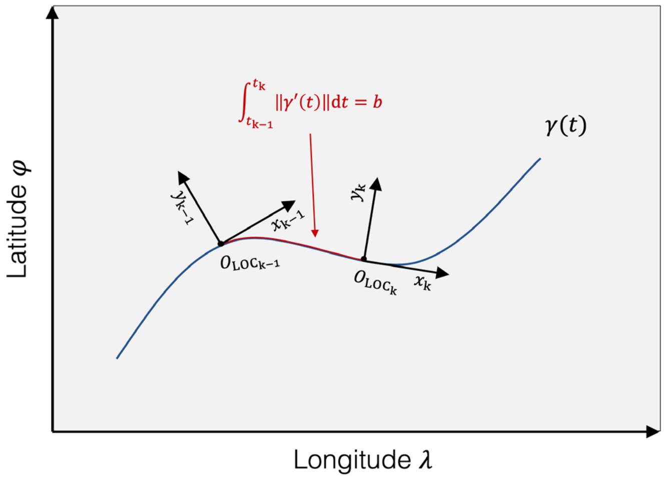


Fig. 3.3. Local reference systems $\{\mathrm{LOC}_{k-1}\}$ and $\{\mathrm{LOC}_k\}$ for vineyard subsets $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$ definition, with origin in $\gamma(t_{k-1}) = \boldsymbol{O}_{\mathrm{LOC}_{k-1}}$ and $\gamma(t_k) = \boldsymbol{O}_{\mathrm{LOC}_k}$, respectively.

Vine row section $\mathcal{S}_k^{\{\mathrm{LOC}_k\}}$ can thus be defined as the following subset of point cloud $\mathcal{PC}^{\{\mathrm{LOC}_k\}}$ (represented in the local reference frame), as follows

$$\mathcal{S}_k^{\{\mathrm{LOC}_k\}} = \left\{ [x, y, z]^{\mathrm{T}} \in \mathcal{PC}^{\{\mathrm{LOC}_k\}} \,\middle|\, |x| \leq \frac{a}{2},\ |y| \leq \frac{\delta_k}{2} \right\} \tag{3}$$

where a and δ_k are the dimensions (m) of $\mathcal{S}_k$ along the x_k and y_k axes, respectively. Please note that a represents a model parameter to be properly tuned. Indeed, a can generally assume different values within a limited range, which should at the same time guarantee a minimum value of $\mathrm{card}(\mathcal{S}_k^{\{\mathrm{LOC}_k\}})$ (lower limit), and allow the vine-row section to be considered as rectilinear (upper limit). A sample of subset $\mathcal{S}_{268}$ centred in $\boldsymbol{O}_{\mathrm{LOC}_{268}}^{\{\mathrm{WGS84}\}} = [44.62447°\ 8.00105°\ 364.6\ \mathrm{m}]^{\mathrm{T}}$ is shown in Figs 3.2 and 3.4, selected with $\vartheta_{268} = 63.4°$, $\delta_{268} = 2.6$ m and $a = 0.8$ m. Henceforth, only the local metric Cartesian reference frame will be used, and thus its explicit dependence from $\{\mathrm{LOC_k}\}$ will be omitted.

3.2.2 Semantic interpretation for vine canopy detection

Once subset $\mathcal{S}_{\mathrm{k}}$ is selected, the next step consists in automatically detecting the set of points $\mathcal{C}_{\mathrm{k}}$ representing the canopy, distinguishing it from those representing the inter-row terrain. Since the terrain elevation of two adjacent inter rows may differ in vineyards located in hilly regions, the classification is performed by individually considering each side of the vine row $\mathcal{S}_{\mathrm{k}}^{+}$ and $\mathcal{S}_{\mathrm{k}}^{-}$ (Fig. 3.4). Being the origin of the reference system $\{\mathrm{LOC_k}\}$ located in the centre line of the canopy width, $\mathcal{S}_{\mathrm{k}}^{+}$ and $\mathcal{S}_{\mathrm{k}}^{-}$ can easily be defined as

$$\mathcal{S}_{\mathrm{k}}^{+} = \{[x, y, z]^{T} \in \mathcal{S}_{\mathrm{k}} \mid y \geq 0\} \tag{4}$$

and

$$\mathcal{S}_{\mathrm{k}}^{-} = \{[x, y, z]^{T} \in \mathcal{S}_{\mathrm{k}} \mid y < 0\} \tag{5}$$

Focusing on side $\mathcal{S}_{\mathrm{k}}^{+}$ of wine row section $\mathcal{S}_{\mathrm{k}}$, the classification was obtained by determining a plane $\wp_{\mathrm{k}}^{+}$ representing the boundary of the two regions containing respectively the points representing the terrain and those representing the canopy, (Fig. 3.4f). Plane $\wp_{\mathrm{k}}^{+}$ was defined as the plane passing through the two lines parallel to the x_{k} axis

$$L'_{\mathrm{k}} = \{[x, y, z]^{\mathrm{T}} \in \mathbb{R}^{3} \mid y = 0,\ z = 0\} \tag{6}$$

and

$$L''_{\mathrm{k}} = \{[x, y, z]^{\mathrm{T}} \in \mathbb{R}^{3} \mid y = w_{\mathrm{k}}^{+},\ z = h_{\mathrm{k}}^{+}\} \tag{7}$$

where w_k is related to the location along the $\mathrm{y_k}$ axis of the external surface of the canopy wall and h_k is the inter-row path terrain elevation along the $\mathrm{z_k}$ axis (Fig. 3.4c). The value of w_k was determined by the robust peak detection (Mathworks, 2020a, Natick, USA) in the normalised frequencies distribution histogram of points p_i along the y_k axis

$$H_\mathrm{y}(\mathcal{S}_\mathrm{k}^{+}, s) = \mathrm{card}\{p_\mathrm{i} = [x, y, z]^\mathrm{T} \in \mathcal{S}_\mathrm{k}^{+}: |y - s_\mathrm{y}| < \frac{\delta_\mathrm{s}}{2}\} \cdot \mathrm{card}(\mathcal{S}_\mathrm{k}^{+})^{-1} \tag{8}$$

where $s_\mathrm{y} \in Y_\mathrm{k} = \{0, \delta_\mathrm{s}, 2\delta_\mathrm{s}, \dots, y_\mathrm{k,max}\}$, Y_k is the set of all the histogram bins, δ_s is the bin width and $y_\mathrm{k,max}$ is the highest value of the considered y coordinates (Fig. 3.4e). Analogously, the value of h_k is the peak of the normalised frequencies distribution histogram

$$H_\mathrm{z}(\mathcal{S}_\mathrm{k}^{+}, t) = \mathrm{card}\{p_\mathrm{i} = [x, y, z]^\mathrm{T} \in \mathcal{S}_\mathrm{k}^{+}: |z - s_\mathrm{z}| < \frac{\delta_\mathrm{s}}{2}\} \cdot \mathrm{card}(\mathcal{S}_\mathrm{k}^{+})^{-1} \tag{9}$$

where $s_\mathrm{z} \in Z_\mathrm{k} = \{0, \delta_\mathrm{s}, 2\delta_\mathrm{s}, \dots, z_\mathrm{k,max}\}$, Z_k is the set of all the histogram bins and $z_\mathrm{k,max}$ is the highest value of the considered z coordinates (Fig. 3.4d). In Fig. 3.4f, plane $\wp_{268}^{+}$, defined by line L''_{268} with $w_{268} = 0.22$ and $h_{268} = 0.86$, is displayed. Point cloud subset $\mathcal{C}_\mathrm{k}^{+}$ representing the canopy wall of the considered side of vine row section $\mathcal{S}_\mathrm{k}^{+}$ can be thus determined as

$$\mathcal{C}_\mathrm{k}^{+} = \left\{[x, y, z]^T \in \mathcal{S}_\mathrm{k}^{+} \;\middle|\; z \geq \frac{h_\mathrm{k}}{w_\mathrm{k}} y\right\} \tag{10}$$

Performing this procedure to both subsets $\mathcal{S}_\mathrm{k}^{+}$ and $\mathcal{S}_\mathrm{k}^{-}$, the set of all points representing canopy $\mathcal{C}_\mathrm{k}$ for the k[th] section can be obtained by the union of sets $\mathcal{C}_\mathrm{k}^{+}$ and $\mathcal{C}_\mathrm{k}^{-}$, that is $\mathcal{C}_\mathrm{k} = \mathcal{C}_\mathrm{k}^{-} \cup \mathcal{C}_\mathrm{k}^{+}$, with $N_{\mathcal{C}_\mathrm{k}} = \mathrm{card}(\mathcal{C}_\mathrm{k})$.

The results of this clustering procedure for canopy detection, obtained by processing sample subset $\mathcal{S}_{268}^{+}$ and the whole point cloud $\mathcal{PC}^{\{\mathrm{WGS84}\}}$, are shown in Fig. 3.4f and Fig. 3.2b, respectively.

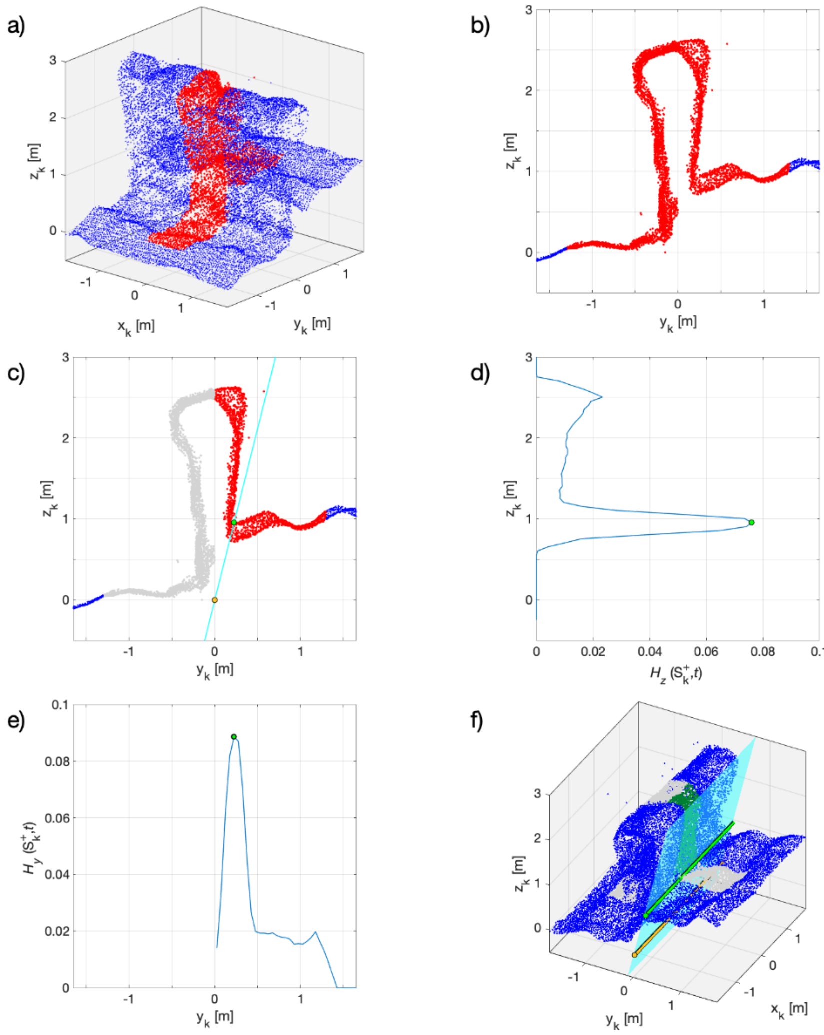


Fig. 3.4. (a) 3D view and (b) 2D view of the sample subset $\mathcal{S}_{268}^{\{\mathrm{LOC_k}\}}$ (red dots) of $\mathcal{PC}^{\{\mathrm{WGS84}\}}$ (blue dots), located in $O_{\mathrm{LOC}_{268}}^{\{\mathrm{WGS84}\}} = [44.62447°\ 8.00105°\ 364.6\ \mathrm{m}]$ and defined with $\vartheta_{268} = 63.4°$, $\delta_{268} = 2.6$ m and $a = 0.8$; (c) 2D view of vine row side $\mathcal{S}_{268}^{+}$ (red), plane $\wp_{268}^{+}$ (light blue) by lines L'_{268} (orange) and L''_{268} (green); (d) normalised frequencies distribution histogram $H_{\mathrm{z}}(\mathcal{S}_{268}^{+}, s)$ and peak location h_{k} (green); (e) normalised frequencies distribution histogram $H_{\mathrm{y}}(\mathcal{S}_{268}^{+}, s)$ (red) and peak location w_{268} (green); and (f) 3D view of detected

canopy cluster $\mathcal{C}_{268}^{+}$ (green dots), plane $\wp_{268}^{+}$ (light blue), lines L'_{268} (orange) and L''_{268} (green).

3.2.3 Canopy model simplification

In this section, the processing step aimed at reducing the complexity (and density) of point set $\mathcal{C}_{\mathrm{k}}$ is presented. This is performed by defining a set of few representative points and, finally, by building a triangulated mesh representing the canopy in the k$^{\text{th}}$ vineyard section. For the sake of readability, subscript k referring to the specific section is omitted in this section.

Hence, the problem considered in this section is the following: *given a point cloud $\mathcal{C}$ of cardinality $N_{\mathcal{C}}$, find a simplified representation of it with low complexity*. The formal meaning of "simplified representation" will be made clear below. The simplification consists of two main steps: first, set $\mathcal{C}$ is "filtered-out" from the outliers and, second, an appropriately defined simplified representation of the outlier-filtered set is derived. The idea behind these two procedures is a dimensionality reduction of the problem, achieved by considering the two-dimensional projection of set $\mathcal{C}$ on plane $x = 0$

$$\mathcal{C}^{2\mathrm{D}} = \{[\mathrm{x,y}]^{\mathrm{T}} \in \mathbb{R}^2 \mid \mathrm{x} = y, \mathrm{y} = z, [x, y, z\,]^{\mathrm{T}} \in \mathcal{C}\} \qquad (11)$$

A graphical representation of set $\mathcal{C}_{268}^{2D}$ relative to section $k = 268$ is shown in Fig. 3.5a.

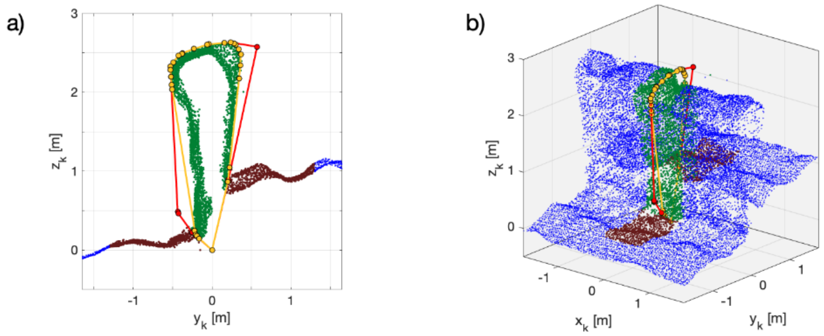


Fig. 3.5. (a) Convex hull polygon $\mathcal{H}[U^{[0]}]$ (red) enclosing all $\mathcal{C}_{268}^{2\mathrm{D}}$ points (green dots) and convex hull polygon $\mathcal{H}[U^{[8]}]$, after removing 8 outliers (orange); and (b) their 3D view.

2.2.3.1 Outlier removal

Examining Fig. 3.5a, it is clear that point set $\mathcal{C}$ contains points which do not properly belong to the canopy. These outliers may be either due to measurement noise and errors, or they may represent artefacts introduced by the algorithm responsible for the point cloud generation. To remove the outlier, a novel technique was proposed, which is based on 2D representation in (Eq. 11) and on the concept of the *convex hull* of a set of point, whose definition is formally recalled next (see e.g. de Berg, van Kreveld, Overmars, Cheong, 2000).

Definition 1. (*Convex hull of a set of points*) Given a point set $\mathcal{C}$, its convex hull is defined as the smallest convex set containing $\mathcal{C}$.

In our case, given $N_{\mathcal{C}}$ two-dimensional points, their convex hull is a convex polygon $\mathcal{H}$ with a number of vertices $N_{\mathcal{H}} \leq N_{\mathcal{C}}$. It should be noted that, while the computation of the convex hull of a set of points in *n* dimensions is in general computationally demanding, in the case of 2D points there exist efficient methods with complexity $O(N_{\mathcal{C}} \log N_{\mathcal{H}})$ – and hence loglinear worst case complexity.

Given 2D set $\mathcal{C}^{2\mathrm{D}}$, its convex hull was denoted as

$$\mathcal{H} = \mathcal{H}[U^{2\mathrm{D}}] = convhull(\,\mathcal{C}^{2\mathrm{D}}) \tag{12}$$

where $U^{2\mathrm{D}} = \{\boldsymbol{u}_{\mathrm{i}} = [\zeta_{\mathrm{i}}, \eta_{\mathrm{i}}]^{\mathrm{T}}, \mathrm{i} = 1, \dots, N_{\mathcal{H}}\}$ are the vertices of the polygon. Recall that, by construction, the vertices of $\mathcal{H}[U^{2\mathrm{D}}]$ represent a subset of the points in $\mathcal{C}^{2\mathrm{D}}$.

Before presenting the algorithm for outlier detection, a result which provides a useful close-form expression for computing the area of a polygon starting from the set of its vertices $U^{2\mathrm{D}}$ is now reported. This formula is termed *Gauss's area formula* or *shoelace formula*, see Boland and Urrutia (2000).

Proposition 1 (*Area of a polygon given its vertices*). Let $\mathcal{H}[U^{2\mathrm{D}}]$ be a polygon of vertices $U^{2\mathrm{D}} = \{\boldsymbol{u}_{\mathrm{i}} = [\zeta_{\mathrm{i}}, \eta_{\mathrm{i}}]^{\mathrm{T}}, \mathrm{i} = 1, \dots, N_{\mathcal{H}}\}$. The two-dimensional Lebesgue measure ($Area$) of $\mathcal{H}[U^{2\mathrm{D}}]$ may be computed as

$$\begin{aligned} &Area(\mathcal{H}[U^{2\mathrm{D}}]) \\ &= \frac{1}{2}\left[\left(\sum_{i=1}^{N_{\mathcal{H}}-1} \left|\zeta_{\mathrm{i}}\eta_{\mathrm{i}+1} + \zeta_{N_{\mathcal{H}}}\eta_{\mathrm{i}}\right|\right)\right. \\ &\left. - \left(\sum_{\mathrm{i}=1}^{N_{\mathcal{H}}-1} \left|\zeta_{\mathrm{i}+1}\eta_{\mathrm{i}} + \zeta_{\mathrm{i}}\eta_{N_{\mathcal{H}}}\right|\right)\right]. \end{aligned} \tag{13}$$

The idea behind the proposed method for outlier detection is as follows: 1) the convex hull of point set $\mathcal{C}$ was constructed, and thus its vertices $U^{2\mathrm{D}}$ determined (these are also points in $\mathcal{C}^{2\mathrm{D}}$); 2) the vertices of $\mathcal{H}[U^{2\mathrm{D}}]$ were removed one-by-one from $\mathcal{C}^{2\mathrm{D}}$, and the area of the remaining set was computed; 3) the vertex which provides the larger area reduction was selected as outlier. This method is formally described in the next algorithm.

Algorithm 1 (*Outlier removal*)

<u>Input</u>: 2D point set $\mathcal{C}^{2\mathrm{D}}$

<u>Output</u>: an outlier-filtered 2D point set $\tilde{\mathcal{C}}^{2\mathrm{D}}$

0. Let $\mathrm{j} = 0$ and set $\mathcal{C}^{[0]} = \mathcal{C}^{2\mathrm{D}}$
1. Compute $\mathcal{H}[U^{[\mathrm{j}]}] = convhull(\mathcal{C}^{[\mathrm{j}]})$
2. For $\ell = 1$ to card($U^{[\mathrm{j}]}$)
    a. Compute $\mathcal{P}[U^{[\mathrm{j}]}_{\backslash \ell}] = convhull(\mathcal{C}^{[\mathrm{j}]}_{\backslash \ell})$, with $\mathcal{C}^{[\mathrm{j}]}_{\backslash \ell} = \mathcal{C}^{[\mathrm{j}]} \backslash \{v_\ell\}$
3. Let $\mathcal{C}^{[\mathrm{j}+1]} = \mathcal{C}^{[\mathrm{j}]}_{\backslash \ell^*}$ with $\ell^* = \arg\min_{\ell} Area(\mathcal{H}[U^{[\mathrm{j}]}_{\backslash \ell}])$
4. If EXITCOND return $\tilde{\mathcal{C}}^{2\mathrm{D}} = \mathcal{C}^{[\mathrm{j}+1]}$, else let $\mathrm{j} = \mathrm{j} + 1$ and go to 1.

A few comments are in order regarding Algorithm 1. First, the condition EXITCOND can easily be set by imposing a desired number of outliers to be removed. However, a better condition is usually provided by considering the area reduction at step j (given by $\Delta A^{[\mathrm{j}]} = Area(\mathcal{C}^{[\mathrm{j}]}_{\backslash \ell^*}) - Area(\mathcal{C}^{[\mathrm{j}-1]}_{\backslash \ell^*})$). When this reduction is below a given threshold, it was interpreted by the fact that the removed point is indeed not an outlier. Second, the computational complexity of the algorithm is polynomial in the cardinality of $\mathcal{C}^{2\mathrm{D}}$, since at each step it requires the computation of card($U^{[\mathrm{j}]}$) convex hulls. The worst possible case is when all points of $\mathcal{C}^{2\mathrm{D}}$ belong to the convex hull (e.g. point on a circumference): in this case, the complexity of removing one outlier is of the order $O(N_{\mathcal{C}}^2 \log N_{\mathcal{C}})$. Some steps of the procedure of outlier removal are shown in Fig. 3.5, where a set of convex hull $\mathcal{H}[U]$ are shown for the processing of $\mathcal{C}^{2\mathrm{D}}_{268}$.

3.2.3.2 Polygonal approximation

To approximate the outlier-filtered 2D point set $\tilde{\mathcal{C}}^{2\mathrm{D}}$, the concept of *c*-gon was introduced. In words, a *c*-gon is a *polygon with exactly c vertices*.

Definition 2. (*c-gon*). A *c*-gon $\mathcal{P}^c = \mathcal{P}[V^{2\mathrm{D}}]$ is defined as a two dimensional polytope (polygon) with c vertices

$$V^{2\mathrm{D}} = \{\boldsymbol{v}_\mathrm{i} = [\zeta_\mathrm{i}, \eta_\mathrm{i}]^\mathrm{T}, \mathrm{i} = 1, \dots, c\} \tag{14}$$

The vertices are assumed to be *ordered in a counter-clockwise way*. An example of *c*-gon $\mathcal{P}^c = \mathcal{P}[V^{2\mathrm{D}}]$ is given in Fig. 3.6a, with $c = 7$.

The following optimisation problem was then formulated:

Problem 1 (*Minimum area c-gon containing a point set*). *Given a point set* $\mathcal{C}^{2\mathrm{D}} = \left\{\boldsymbol{p}_\mathrm{i} = \left[\mathrm{x}_\mathrm{i}, \mathrm{y}_\mathrm{i}\right]^\mathrm{T} \in \mathbb{R}^2, \mathrm{i} = 1, \dots, N_\mathcal{C}\right\}$, *find the c-gon* $\mathcal{P}^c = \mathcal{P}[V^{2\mathrm{D}}]$ *of minimum area such that* $\mathcal{C}^{2\mathrm{D}} \subseteq \mathcal{P}^c$. This is formulated as follows:

$$\begin{aligned} \mathcal{P}[V^{2\mathrm{D}*}] &= \arg\min_{V^{2\mathrm{D}}} Area(\mathcal{P}[V^{2\mathrm{D}}]) \\ s.t.\ \ \boldsymbol{p}_\mathrm{i} &\in \mathcal{P}[V^{2\mathrm{D}}], \mathrm{i} = 1, \dots, N_\mathcal{C} \end{aligned} \tag{15}$$

Theorem 1 (*Minimum enclosing c-gon as bilinear program*). The solution to the minimum area *c*-gon enclosing a given set of points $\mathcal{C}^{2\mathrm{D}}$ can be found as the solution of the following bilinear program

$$\begin{aligned} (\boldsymbol{\zeta}^*, \boldsymbol{\eta}^*) &= \arg\min_{(\boldsymbol{\zeta},\boldsymbol{\eta})} \boldsymbol{\zeta}^\mathrm{T} \boldsymbol{S} \boldsymbol{\eta} \\ \text{s.t. } [\zeta_\mathrm{j} \quad \zeta_{\mathrm{j}+1}] \boldsymbol{D} \begin{bmatrix} \eta_\mathrm{j} \\ \eta_{\mathrm{j}+1} \end{bmatrix} &+ \boldsymbol{d}^\mathrm{T} \mathrm{y}_\mathrm{i} \begin{bmatrix} \zeta_\mathrm{j} \\ \zeta_{\mathrm{j}+1} \end{bmatrix} - \boldsymbol{d}^\mathrm{T} \mathrm{x}_\mathrm{i} \begin{bmatrix} \eta_\mathrm{j} \\ \eta_{\mathrm{j}+1} \end{bmatrix} \leq 0, \\ &\mathrm{j} = 1, \dots, c-1 \quad \mathrm{i} = 1, \dots, N_\mathcal{C} \\ [\zeta_c \quad \zeta_1] \boldsymbol{D} \begin{bmatrix} \eta_c \\ \eta_1 \end{bmatrix} &+ \boldsymbol{d}^\mathrm{T} \mathrm{y}_\mathrm{i} \begin{bmatrix} \zeta_c \\ \zeta_1 \end{bmatrix} - \boldsymbol{d}^\mathrm{T} \mathrm{x}_\mathrm{i} \begin{bmatrix} \eta_c \\ \eta_1 \end{bmatrix} \leq 0, \\ &\mathrm{i} = 1, \dots, N_\mathcal{C} \end{aligned} \tag{16}$$

where $\boldsymbol{\zeta} = (\xi_1 \cdots \xi_c)^T$, $\boldsymbol{\eta} = (\eta_1 \cdots \eta_c)^T$, and $\boldsymbol{S} = \tilde{\boldsymbol{S}} - \tilde{\boldsymbol{S}}^\mathrm{T}$, with

$$\tilde{\boldsymbol{S}} = \left[\left[\begin{pmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}\right]\right], \boldsymbol{D} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \boldsymbol{d} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{17}$$

Before sharing proof of the above result, a few considerations should be made: first, that Eq. (16) was indeed noted to be a bilinear problem, as cost $\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{\eta}$ is bilinear (note that matrix $\boldsymbol{S}$ is skew-symmetric by construction), and also that the constraints are bilinear equations of variables $(\boldsymbol{\zeta}, \boldsymbol{\eta})$.

Proof of Theorem 1.

By applying Eq. (13), the cost function in Eq. (16) is immediately rewritten as

$$\begin{aligned} Area(\mathcal{P}[V^c]) &= \frac{1}{2}\left[\left(\sum_{\mathrm{i}=1}^{c-1} \zeta_{\mathrm{i}}\eta_{\mathrm{i}+1} + \zeta_{\mathrm{n}}\eta_{\mathrm{i}}\right) - \left(\sum_{\mathrm{i}=1}^{c-1} \zeta_{\mathrm{i}+1}\eta_{\mathrm{i}} + \zeta_{\mathrm{i}}\eta_{\mathrm{n}}\right)\right. \\ &= \frac{1}{2}\left[\left(\boldsymbol{\zeta}^{\mathrm{T}}\tilde{\boldsymbol{S}}\boldsymbol{\eta}\right) - \left(\boldsymbol{\eta}^{\mathrm{T}}\tilde{\boldsymbol{S}}\boldsymbol{\zeta}\right)\right] \\ &= \frac{1}{2}\left[\left(\boldsymbol{\zeta}^{\mathrm{T}}\tilde{\boldsymbol{S}}\boldsymbol{\eta}\right) - \left(\boldsymbol{\zeta}^{\mathrm{T}}\tilde{\boldsymbol{S}}^{\mathbf{T}}\boldsymbol{\eta}\right)\right] = \frac{1}{2}\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{\eta} \end{aligned} \tag{18}$$

The cost in Eq. (16) follows immediately by noticing that constant ½ is irrelevant for the optimization problem. The constraints in Eq. (16) are harder to derive.

To impose that the point $\boldsymbol{p}_{\mathrm{i}}$ is contained in the c-gon $\mathcal{P}[V^{2\mathrm{D}}]$, it must lie on the left of the vector $(\boldsymbol{v}_{\mathrm{j}+1} - \boldsymbol{v}_{\mathrm{j}})$, for all j (see Fig. 3.6b). This is equivalent to imposing that the sign of the cross (external) product of vector $(\boldsymbol{v}_{\mathrm{j}+1} - \boldsymbol{v}_{\mathrm{j}})$ with vector $(\boldsymbol{p}_{\mathrm{i}} - \boldsymbol{v}_{\mathrm{j}})$ is negative, i.e.

$$(\boldsymbol{v}_{\mathrm{j}+1} - \boldsymbol{v}_{\mathrm{j}}) \times (\boldsymbol{p}_{\mathrm{i}} - \boldsymbol{v}_{\mathrm{j}}) = (\mathrm{x}_{\mathrm{i}} - \zeta_{\mathrm{j}})(\eta_{\mathrm{j}+1} - \eta_{\mathrm{j}}) - (\zeta_{\mathrm{j}+1} - \zeta_{\mathrm{j}})(\mathrm{y}_{\mathrm{i}} - \eta_{\mathrm{j}}) \leq 0 \tag{19}$$

The proof is completed by realizing that this equation immediately rewrites as the first constraint in Eq. (16) by introducing the quantities ***D*** and ***d***. This equation should hold for all points $\boldsymbol{p}_{\mathrm{i}}\ \mathrm{i}=1,\dots,N_{\mathcal{C}}$, and for all couples of vertices $\boldsymbol{v}_{\mathrm{j}},\boldsymbol{v}_{\mathrm{j}+1}$, $\mathrm{j}=1,\dots,c-1$. The last equation takes into account the line passing through the two vertices $\boldsymbol{v}_c,\boldsymbol{v}_1$. Since Eq. (16) was found to be bilinear and, hence, nonconvex, it generally presents potential local minima. However, rather efficient algorithms exist for this specific class of problems. To obtain a more accurate canopy model and to speed up this bilinear problem solution, the three lower points of the *c*-gon were considered fixed (Fig. 3.6c), always in position $\boldsymbol{v}_1=[0\quad 0]$, $\boldsymbol{v}_2=[w^+\quad h^+]$ and $\boldsymbol{v}_c=[w^-\quad h^-]$, allowing to remove the last constraint in Eq. (15), which would be automatically satisfied.

Finally, the determined vertices V^{2D} of polygon $\mathcal{P}[V^{\mathrm{2D}}]$ are represented in the original 3D reference system $\{\mathrm{LOC_k}\}$ as

$$V_{\mathrm{k}}^{\{\mathrm{LOC_k}\}}=\{\boldsymbol{v}_{\mathrm{i}}=[0,\zeta_{\mathrm{i}},\eta_{\mathrm{i}}]^{\mathrm{T}}\mid \boldsymbol{v}_{\mathrm{i}}=[\zeta_{\mathrm{i}},\eta_{\mathrm{i}}]^{\mathrm{T}}\in V^{\mathrm{2D}}\} \qquad (20)$$

and then in the absolute $\{\mathrm{WGS84}\}$ in order to make them suitable for the final processing step to determine a low complexity triangulated mesh generation. In Fig. 3.6c, a *c*-gon $\mathcal{P}^7$ enclosing the given set of points $\mathcal{C}_{268}^{\mathrm{2D}}$ with vertices V_{268}^{2D} is represented, with fixed vertices $\boldsymbol{v}_1=[0\ 0]$, $\boldsymbol{v}_2=[w_{268}^+\quad h_{268}^+]=[0.22\quad 0.87]$ and $\boldsymbol{v}_7=[w_{268}^-\quad h_{268}^-]=[-0.28\quad 0.26]$, whereas its representation in the 3D reference system $\{\mathrm{LOC_k}\}$ can be observed in Fig. 3.6d.

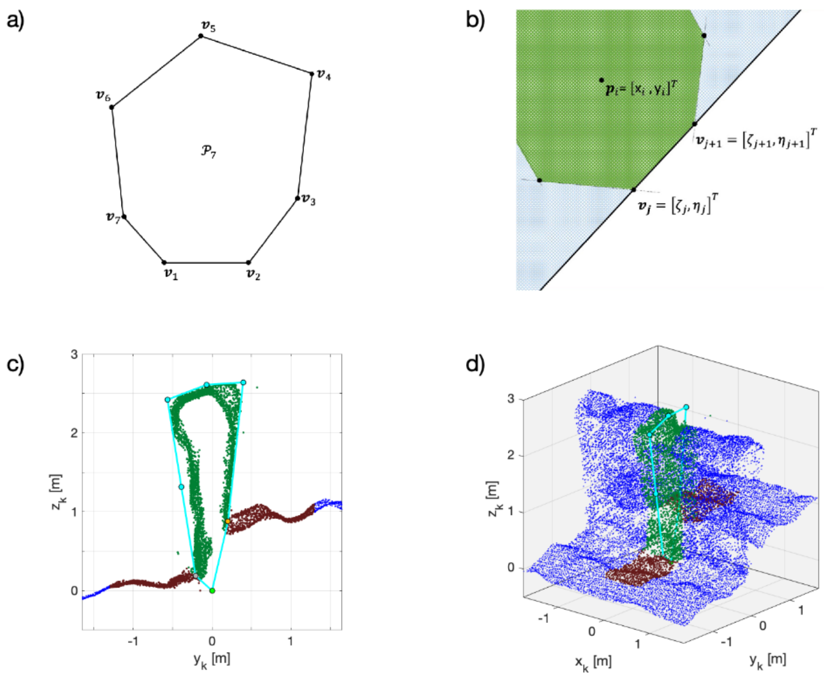


Fig. 3.6. (a) $c - gon$ with $c = 7$ vertices numbered in a counter-clockwise direction; (b) the point p_i contained in the c-gon (green area): a point is contained in the c-gon if it lies on the left of the vector $(v_{j+1} - v_j)$, for all j (light blue area); (c) c-gon $\mathcal{P}^7$ enclosing the given set of points $\mathcal{C}_{268}^{2D}$ (cyan line) with vertices V_{268}^{2D} (of which $\boldsymbol{v}_1 = [0\ 0]$ (green dot), $\boldsymbol{v}_2 = [w_{268}^{+} \quad h_{268}^{+}] = [0.22 \quad 0.87]$ (orange dot) and $\boldsymbol{v}_7 = [w_{268}^{-} \quad h_{268}^{-}] = [-0.28 \quad 0.26]$ (grey dot) are fixed vertices); (d) 3D view of minimum area c-gon enclosing the given set of points $\mathcal{C}_{268}$ with vertices in $V_{268}^{\{\mathrm{LOC}_{268}\}}$.

2.2.3.3 Triangulated mesh building

The low complexity model of the canopy is defined as a triangulated mesh

$$\mathcal{M}_{\mathrm{k}}^{[a,b,c]} = \left[\{V_{\mathrm{k}-1}, V_{\mathrm{k}}\}, F_{\mathrm{k}-1,\mathrm{k}}\right] \tag{21}$$

where $V_{\mathrm{k}-1}$ and V_{k} are the sets of mesh vertices, described in the previous section, and $F_{\mathrm{k}-1,\mathrm{k}}$ is the set of triangular faces of the mesh between them (Fig. 3.7b). A triangular face is defined as triplets of points $\mathfrak{v}$, so that $F_{\mathrm{k-1,k}}$ can be expressed as

$$F_{\mathrm{k-1,k+1}} = \{(\mathfrak{v}_{\mathrm{k-1,i}}, \mathfrak{v}_{\mathrm{k-1,i+1}}, \mathfrak{v}_{\mathrm{k,i}}), (\mathfrak{v}_{\mathrm{k-1,i+1}}, \mathfrak{v}_{\mathrm{k,i+1}}, \mathfrak{v}_{\mathrm{k,i}})\ \forall\ \mathrm{i} = 1, \dots, c\} \tag{22}$$

A graphical representation of a low complexity triangulated mesh model $\mathcal{M}$ obtained by processing two consecutive polygons $\mathcal{P}[V_{268}]$ and $\mathcal{P}[V_{269}]$, having model parameters $a = 0.8\ m$, $b = 0.5\ m$ and $c = 7$, can be observed in Fig. 3.7b. A sample portion of raw 3D point cloud $\mathcal{PC}^{\{\mathrm{WGS84}\}}$ (blue dots) and low complexity triangulated 3D mesh model $\mathcal{M}$, generated by linking polygon vertices $\mathcal{P}[V_{\mathrm{k}}]$ between adjacent sections $\mathcal{S}_{\mathrm{k}}$ can be observed in Figs 3.2a and 3.2c, respectively. The procedure described in the previous sections, was repeated along the vine row model for all vine row section $\mathcal{S}_{\mathrm{k}}$ with $\mathrm{k} \in \mathcal{K}$.

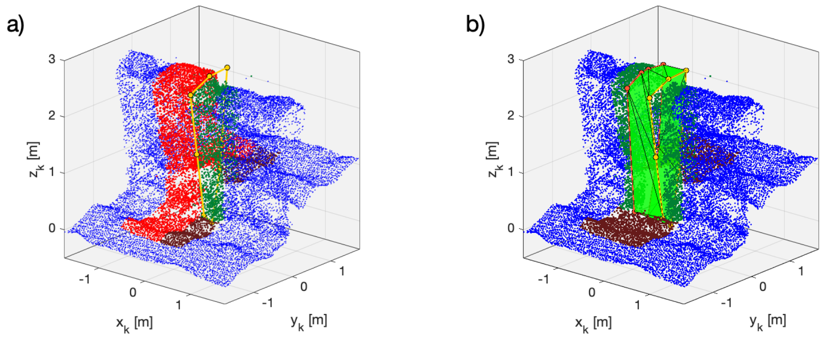


Fig. 3.7. Low complexity triangulated mesh generation: (a) c-gons $\mathcal{P}[V_{268}^{\{\mathrm{LOC}_{268}\}}]$ (cyan) and $\mathcal{P}[V_{267}^{\{\mathrm{LOC}_{267}\}}]$ vertices (red) and (b) the generated low complexity mesh $\mathcal{M}_{268}$.

3.3. Results and discussion

The low complexity triangulated 3D mesh model $\mathcal{M}^{[a,b,c]}$ of vineyards was strictly related to 3 main parameters: (1) the width a of sections $\mathcal{S}$,

(2) the distance b between two adjacent sections $\mathcal{S}_{\mathrm{k}}$ and $\mathcal{S}_{\mathrm{k+1}}$ and (3) the number of points c used to properly describe every vine row section. The effect of different choices of these parameters on the final mesh model $\mathcal{M}^{[a,b,c]}$ layout were multiple and linked: the a parameter affects the average amount of points that were considered in a section $\mathcal{S}$ which, together with the c parameter, conditions the *c-gon* $\mathcal{P}[V^{2D}]$ shape; this final aspect, joined with the effect of parameter c, affected the accuracy of the mesh in modelling the canopy of the vineyard. Depending on the values of these three parameters, the quality of the computed 3D mesh model can thus vary considerably. The optimal configuration of the modelling framework was determined by an optimal search process via a genetic algorithm, based on the quality score Q of mesh model $\mathcal{M}^{[a,b,c]}$. Parameters $[a\ b\ c]$ were varied within ranges [0.1, 1] m, [0.1, 2] m and [5, 11], respectively. The quality scoring function Q was evaluated by comparing the generated 3D mesh model $\mathcal{M}^{[a,b,c]}$ to the raw, highly detailed, point cloud section $\mathcal{C}$, and defined as

$$Q_{\mathrm{k}} = G_{\mathrm{k}} - (U_{\mathrm{k}} + O_{\mathrm{k}}) + R_{\mathrm{k}} \tag{23}$$

where G_{k} is the *good-modelling* index, U_{k} and O_{k} are the two *under-modelling* and *over-modelling* error indices, respectively, and R_{k} is the *complexity reduction* index. The description and definition of these four indices are presented in Table 3.1, where A_g, A_u and A_o are the areas derived from the intersection of $\mathcal{M}_{\mathrm{k}}$ with a plane perpendicular to line γ (Fig. 3.8) and where V_{ref} is the $\mathcal{C}_{\mathrm{k}}$ envelope volume.

Table 3.1. Indexes for quality score $\mathcal{Q}$ computation of mesh model $\mathcal{M}$.

Name	Description	Definition
good-modelling index G_{k}	volume of $\mathcal{C}_{\mathrm{k}}$ properly modelled in $\mathcal{M}_{\mathrm{k}}$	$G_{\mathrm{k}} = V_{\mathrm{ref}}^{-1} \cdot \int_{t_{\mathrm{k}}}^{t_{\mathrm{k+1}}} A_g\big(\gamma(t)\big)\lVert\gamma'(t)\rVert \mathrm{d}t$
under-modelling index U_{k}	volume of $\mathcal{C}_{\mathrm{k}}$ not modelled in $\mathcal{M}_{\mathrm{k}}$	$U_{\mathrm{k}} = V_{\mathrm{ref}}^{-1} \cdot \int_{t_{\mathrm{k}}}^{t_{\mathrm{k+1}}} A_u\big(\gamma(t)\big)\lVert\gamma'(t)\rVert \mathrm{d}t$
over-modelling index O_{k}	volume of $\mathcal{M}_{\mathrm{k}}$ not present in $\mathcal{C}_{\mathrm{k}}$	$O_{\mathrm{k}} = V_{\mathrm{ref}}^{-1} \cdot \int_{t_{\mathrm{k}}}^{t_{\mathrm{k+1}}} A_o\big(\gamma(t)\big)\lVert\gamma'(t)\rVert \mathrm{d}t$
complexity reduction index R_{k}	storage space reduction of $\mathcal{M}_{\mathrm{k}}$ compared to $\mathcal{C}_{\mathrm{k}}$	$R_{\mathrm{k}} = 10^{-1} \cdot \left(1 - \frac{\mathrm{card}(V_k) - 3 \cdot \mathrm{card}\big(F_{\mathrm{k-1,k}}\big)}{\mathrm{card}(\mathcal{C}_{\mathrm{k}}) - \mathrm{card}(\mathcal{C}_{\mathrm{k}} \cap \mathcal{C}_{\mathrm{k-1}})}\right)$

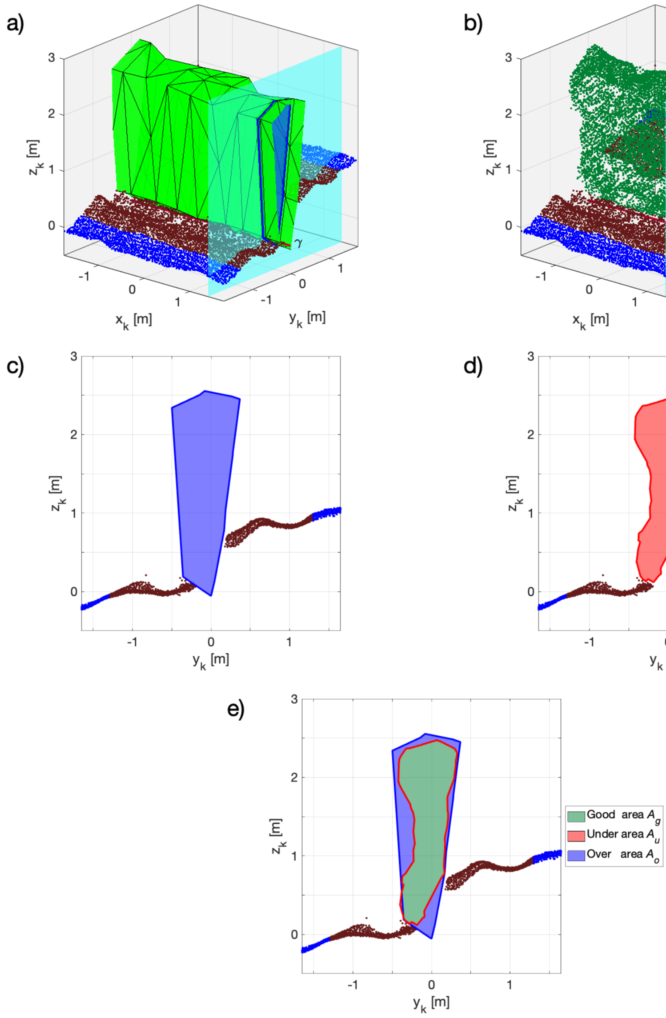


Fig. 3.8. Model quality indices evaluation: (a-c) model $\mathcal{M}_\mathrm{k}$ section; (b-d) points $\mathcal{C}_\mathrm{k}$ envelope on line γ perpendicular plane; and (e) areas A_g, A_u and A_o, determined by comparing the two polygon in (c-d).

In order to detect the optimal configuration of the defined modelling framework and to validate it, the procedure, discussed in section 2, was implemented in the Matlab® environment (Mathworks, 2020b, Natick, USA) and a point cloud of more than 128 m of vine rows was processed. Depending on the model parameter values, the overall number of processed vine row sections ranged from 1,280 for models $\mathcal{M}$ with $b = 0.25$ m to 64 for those with $b = 2$ m.

The results of the optimisation process, performed by the ant colony genetic algorithm (Mathworks, 2020c, Natick, USA), showed that model $\mathcal{M}^{[0.4,0.25,7]}$ (with a =0.4 m, b =0.25 m and c =7) obtained the highest average quality score, which was $\bar{Q} = 0.71$, and a standard deviation of $\sigma_Q = 0.19$ (Fig. 3.9e). More in detail, considering the best model $\mathcal{M}^{[0.4,0.25,7]}$, the histograms of the indices G_{k}, U_{k}, O_{k} and R_{k} values, assessed on all the 496 considered vine row sections $\mathcal{S}_{\mathrm{k}}$, are reported in Fig. 3.9. The *good modelling* index G_{k} had an overall mean value of $\bar{G}_{\mathrm{m}} = 0.92$ and a standard deviation of $\sigma_G = 0.07$. The indices describing errors in modelling the canopy produced low values, with mean indices of under U_{k} and over O_{k} modelling equal to $\bar{U}_{\mathrm{m}} = 0.07$ and $\bar{O}_{\mathrm{m}} = 0.23$, having a standard deviation of $\sigma_U = 0.07$ and $\sigma_O = 0.14$, respectively. Finally, the complexity reduction index R_{k} had a mean of $\bar{R}_{\mathrm{m}} = 0.09$ and a very small standard deviation of $\sigma_R = 0.05 \cdot 10^{-2}$.

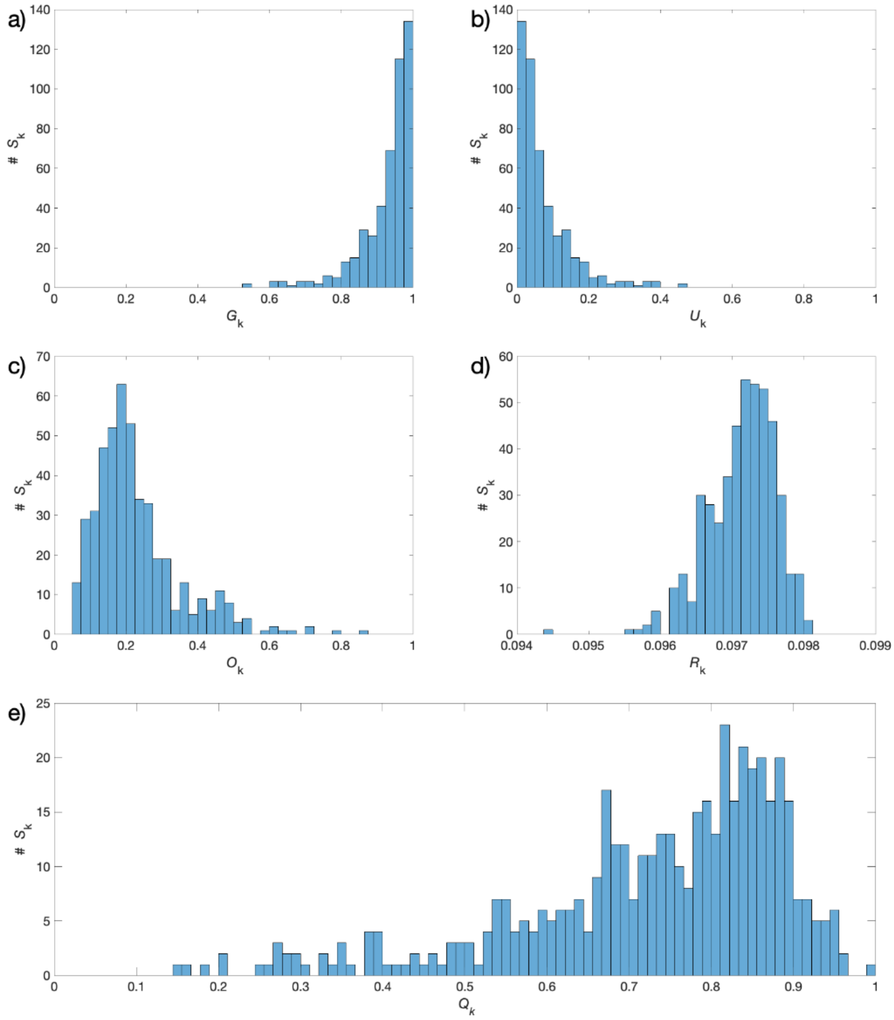


Fig. 3.9. Model quality indices results histogram obtained by the model $\mathcal{M}^{[0.4,0.25,7]}$ (with a =0.4 m, b =0.25 m and c =7): (a) *good modelling* G_{k}; (b) *under modelling* U_{k}; (c) *over modelling* O_{k}; (d) *complexity reduction* R_{k} indices; and (e) *quality score* Q_{k}.

As can be noted from the obtained results, the proposed modelling framework achieved a very high *good-modelling* index and very low *under-modelling* index, which confirmed the reliability of the modelled canopy volumes. Indeed, the slightly higher values obtained for the over-detection index are related to the specifically adopted approach, which is aimed at providing a robust and precautionary low complexity canopy envelope. This solution guarantees, for example, the risk reduction of collisions with vines when simplified 3D meshes are used for UGV path planning.

The modelled vineyard dataset turned out to be more than 98% “lighter” compared to the original point clouds dataset, while assuring minimal loss of canopy shape information. A low complexity triangulated 3D mesh model $\mathcal{M}$ of a portion of raw 3D point cloud $\mathcal{PC}^{\{\mathrm{WGS84}\}}$ consisting of 4 vine rows, processed for the best model $\mathcal{M}^{[0.4,0.25,7]}$ parameters (with a =0.4 m, b =0.25 m and c =7) can be observed in Fig. 3.2c.

3.4. Conclusions

An innovative modelling framework has been presented here to generate low complexity 3D mesh models of vine rows from raw 3D point clouds of vineyards. The proposed methodology reduces the amount of georeferenced instances required to properly describe the spatial layout and shape of vine canopies; this allows the amount of data to be drastically reduced without losing relevant crop shape information. In addition, the developed algorithm semantically interprets the 3D model by automatically classifying the points of the could in two groups: one representing the vine canopy and the other terrain.

The optimal configuration of the modelling framework was determined by an optimal search process via a genetic algorithm by varying a set of three relevant modelling parameters, and its effectiveness was investigated by processing more than 128 m of vine rows. For this purpose, a quality score of the generated low complexity triangulated 3D mesh model was evaluated by comparing it with a highly detailed vineyard point cloud. The obtained dataset volume reduction is 98% percent, providing a vineyard low complexity model of about 7 Mb ha^{-1} by processing a vineyard raw point cloud of more than 500 Mb ha^{-1}.

The proposed modelling framework, designed to process 3D point clouds of vineyards cultivated by VSP-training systems, is not hindered by complex scenarios, such as hilly regions and/or non-linear vine rows, as it is able to automatically process non uniform vineyards, in terms of inter- and intra-row distance. The reduction of the amount of data is a crucial factor in facilitating shorter computational times of huge datasets, such as crop raw 3D point clouds, thus enabling the exploitation of point clouds information in real time operations in the field. When considering scenarios involving cooperating machines and robots, data reduction is also relevant for enabling fast communication and data exchange between in field actors.

Funding

This research was partially funded by the project “New technical and operative solutions for the use of drones in Agriculture 4.0” (PRIN 2017, Prot. 2017S559BB).

Acknowledgments

The authors would like to acknowledge the Azienda Agricola Germano Ettore, Serralunga d’Alba, Italy for hosting the experiments and Iway SRL for the support with the drone-based image acquisition.

3.5. References

1. Agisoft©. Accessed November 2019 from http://www.agisoft.com.
2. Bao, Y., Tang, L., Srinivasan, S., Schnable, P.S. (2019). Field-based architectural traits characterisation of maize plant using time-of-flight 3D imaging. *Biosystems Engineering*, 178, 86-101, 10.1016/j.biosystemseng.2018.11.005
3. Bechar, A., Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149, 94-111, 10.1016/j.biosystemseng.2016.06.014
4. Bechar, A., Vigneault, C. (2017). Agricultural robots for field operations. Part 2: Operations and systems. *Biosystems Engineering*, 153, 110-128, 10.1016/j.biosystemseng.2016.11.004
5. Boland, R.P., Urrutia, J. (2000). Polygon Area Problems. *12th Canadian Conference on Computational Geometry* pp. 159-162
6. Campos, J., Llop, J., Gallart, M., Garcia-Ruíz, F., Gras, A., Salcedo, R., et al. (2019). Development of canopy vigour maps using UAV for site-specific management during vineyard spraying process. *Precision Agriculture*, 20, 1136-1156, 10.1007/s11119-019-09643-z
7. Comba, L., Ricauda Aimonino, D., Gay, P. (2016). Robot ensembles for grafting herbaceous crops. *Biosystems Engineering,* 146, 227-239, 10.1016/j.biosystemseng.2016.02.012
8. Comba, L., Biglia, A., Ricauda Aimonino, D., Gay, P. (2018). Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. *Computers and Electronics in Agriculture*, 155, 84-95, 10.1016/j.compag.2018.10.005

9. Comba, L., Biglia, A., Ricauda Aimonino, D., Tortia, C., Mania, E., Guidoni, S., et al. (2019a). Leaf Area Index evaluation in vineyards using 3D point clouds from UAV imagery. *Precision Agriculture*, 10.1007/s11119-019-09699-x
10. Comba, L., Biglia, A., Ricauda Aimonino, D., Barge, P., Tortia, C., Gay, P. (2019b). 2D and 3D data fusion for crop monitoring in precision agriculture. 2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 62-67, 10.1109/MetroAgriFor.2019.8909219
11. Condotta, I.C.F.S., Brown-Brandl, T.M., Pitla, S.K., Stinn, J.P., Silva-Miranda, K.O. (2020). Evaluation of low-cost depth cameras for agricultural applications. *Computers and Electronics in Agriculture*, 173, 105394, 10.1016/j.compag.2020.105394
12. de Berg, M., van Kreveld, M., Overmars, M., Cheong, O. (2000). *Computational Geometry: Algorithms and Applications*, Springer, pp. 2-8
13. Feng, A., Zhou, J., Vories, E.D., Sudduth, K.A., Zhang, M. (2020). Yield estimation in cotton using UAV-based multi-sensor imagery. *Biosystems Engineering*, 193, 101-114, 10.1016/j.biosystemseng.2020.02.014
14. Gebbers, R., Adamchuk, V.I. (2010). Precision Agriculture and Food Security. *Science*, 327, 828-831, 10.1126/science.1183899
15. Grella, M., Gallart, M., Marucco, P., Balsari, P., Gil, E. (2017). Ground deposition and airborne spray drift assessment in vineyard and orchard: The influence of environmental variables and sprayer settings. *Sustainability*, 9, 728, 10.3390/su9050728

16. Grimstad, L., From, P.J. (2017). Thorvald II - a Modular and Re-configurable Agricultural Robot. *IFAC-PapersOnLine*, 50, 4588-4593, 10.1016/j.ifacol.2017.08.1005
17. Hameed, I.A., la Cour-Harbo, A., Osen, O.L. (2016). Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. *Robotics and Autonomous Systems*, 76, 36-45, /10.1016/j.robot.2015.11.009
18. Han, J. (2018). An efficient approach to 3D path planning. *Information Sciences*, 478, 318-330, 10.1016/j.ins.2018.11.045
19. Higgins, S., Schellberg, J., Bailey, J.S. (2019). Improving productivity and increasing the efficiency of soil nutrient management on grassland farms in the UK and Ireland using precision agriculture technology. *European Journal of Agronomy*, 106, pp. 67-74, 10.1016/j.eja.2019.04.001
20. Kassler. M. (2001). Agricultural Automation in the new Millennium. *Comput Electron Agr*, 30, pp. 237-240, 10.1016/S0168-1699(00)00167-8
21. Khaliq, A., Comba, L., Biglia, A., Ricauda Aimonino, D., Chiaberge, M., Gay, P. (2019). Comparison of satellite and UAV-based multispectral imagery for vineyard variability assessment. *Remote Sensing*, 11, 436, 10.3390/rs11040436
22. Luo, L., Tang, Y., Zou, X., Ye, M., Feng, W., Li, G. (2016). Vision-based extraction of spatial information in grape clusters for harvesting robots. *Biosystems Engineering*, 151, 90-104, 10.1016/j.biosystemseng.2016.08.026

23. Mack, J., Lenz, C., Teutrine, J., Steinhage, V. (2017). High-precision 3D detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Comput. Electron. Agric*. 135, pp. 300–311, 10.1016/j.compag.2017.02.017
24. Maes, W.H., Steppe, K. (2019). Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture. *Trends in Plant Science*, 24, 152-164, 10.1016/j.tplants.2018.11.007
25. Mathworks Matlab®, 2020a. Accessed November 2019 from https://uk.mathworks.com/help/signal/examples/peak-analysis.html.
26. Mathworks Matlab®, 2020b. Accessed November 2019 from https://www.mathworks.com/products/matlab.html.
27. Mathworks Matlab®, 2020c. Accessed November 2019 from https://uk.mathworks.com/discovery/genetic-algorithm.html.
28. Miranda-Fuentes, A., Llorens, J., Gamarra-Diezma, J. L., Gil-Ribes, J. A., Gil, E. (2015). Towards an Optimized Method of Olive Tree Crown Volume Measurement. Sensors, 15(2), 3671-3687, 10.3390/s150203671
29. Mortensen, A.K., Bender, A., Whelan, B., Barbour, M.M., Sukkarieh, S., Karstoft, H., et al. (2018). Segmentation of lettuce in coloured 3D point clouds for fresh weight estimation. *Computers and Electronics in Agriculture*, 154, 373-381, 10.1016/j.compag.2018.09.010
30. Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Tona, E., et al. (2016). Selective spraying of grapevines for disease control using a modular agricultural robot. *Biosystems Engineering*, 146, 203-215, 10.1016/j.biosystemseng.2015.12.004

31. Parrot© SA. Accessed November 2019 from https://www.parrot.com/business-solutions-it/professionisti/parrot-sequoia#parrot-sequoia.

32. Pavón-Pulido, N., López-Riquelme, J.A., Torres, R., Morais, R., Pastor, J.A. (2017). New trends in precision agriculture: a novel cloud-based system for enabling data storage and agricultural task planning and automation. *Precision Agriculture*, 18, 1038-1068, 10.1007/s11119-017-9532-7

33. Peng, Y., Xiao, Y., Fu, Z., Dong, Y., Zheng, Y., Yan, H., et al. (2019). Precision irrigation perspectives on the sustainable water-saving of field crop production in China: Water demand prediction and irrigation scheme optimization. *Journal of Cleaner Production*, 230, 365-377, 10.1016/j.jclepro.2019.04.347

34. Primicerio, J., Gay, P., Ricauda Aimonino, D., Comba, L., Matese, A., Di Gennaro, S.F. (2015). NDVI based vigour maps production using automatic detection of vine rows in ultra-high resolution aerial images. European Conference (10th) on Precision Agriculture, 465-470, DOI:10.3920/978-90-8686-814-8_57

35. Primicerio, J., Caruso, G., Comba, L., Crisci, A., Gay, P., Guidoni, S., et al. (2017). Individual plant definition and missing plant characterization in vineyards from high-resolution UAV imagery. *European Journal of Remote Sensing*, 50, 179-186, DOI:10.1080/22797254.2017.1308234

36. Reina, G., Milella, A., Galati, R. (2017). Terrain assessment for precision agriculture using vehicle dynamic modelling. *Biosystems Engineering*, 162, 124-139, 10.1016/j.biosystemseng.2017.06.025

37. Reza, Md.N., Na, I., Baek, S.W., Lee, K.H. (2019). Rice yield estimation based on K-means clustering with graph-cut segmentation using low-altitude UAV images. *Biosystems Engineering*, 177, 109-121, 10.1016/j.biosystemseng.2018.09.014
38. Rosell-Polo, J.R., Gregorio, E., Gené, J., Llorens, J., Torrent, X., Arnó, J., et al. (2017). Kinect v2 sensor-based mobile terrestrial laser scanner for agricultural outdoor applications. *IEEE/ASME Transactions on Mechatronics*, 22, 2420-2427, 10.1109/TMECH.2017.2663436
39. Saberioon, M.M., Cisar, P. (2016). Automated multiple fish tracking in three-Dimension using a Structured Light Sensor. *Computers and Electronics in Agriculture*, 121, 215-221, 10.1016/j.compag.2015.12.014
40. Serazetdinova, L., Garratt, J., Baylis, A., Stergiadis, S., Collison, M., Davis, S. (2019). How should we turn data into decisions in AgriFood?. *Journal of the Science of Food and Agriculture*, 99, 3213-3219, 10.1002/jsfa.9545
41. Sozzi, M., Kayad, A., Giora, D., Sartori, L., Marinello, F. (2019). Cost-effectiveness and performance of optical satellites constellation for Precision Agriculture. European Conference (10th) on Precision Agriculture, 501-507, 10.3920/978-90-8686-888-9_62
42. Tenhunen, H., Pahikkala, T., Nevalainen, O., Teuhola, J., Mattila, H., Tyystjärvi, E. (2019). Automatic detection of cereal rows by means of pattern recognition techniques. *Computers and Electronics in Agriculture*, 162, 677-688, 10.1016/j.compag.2019.05.002
43. Utstumo, T., Urdal, F., Brevik, A., Dørum, J., Netland, J., Overskeid, Ø., et al. (2018). Robotic in-row weed control in vegetables. *Computers*

and Electronics in Agriculture, 154, 36-45, 10.1016/j.compag.2018.08.043

44. Van Evert, F. K., Fountas, S., Jakovetic, D., Crnojevic V., Travlos, I., Kempenaar, C. (2017). Big Data for weed control and crop protection. *Weed research*, 57, 218-233, 10.1111/wre.12255
45. Van Henten, E.J., Bac, C.W., Hemming, J., Edan, Y. (2013). Robotics in protected cultivation. *IFAC Proceedings Volumes*, 46, pp. 170-177, 10.3182/20130828-2-SF-3019.00070
46. Vidoni, R., Bietresato, M., Gasparetto, A., Mazzetto, F. (2015). Evaluation and stability comparison of different vehicle configurations for robotic agricultural operations on side-slopes. *Biosystems Engineering*, 129, 197-211, 10.1016/j.biosystemseng.2014.10.003
47. Wang, L., Wang, P., Liang, S., Qi, X., Li, L., Xu, L. (2019). Monitoring maize growth conditions by training a BP neural network with remotely sensed vegetation temperature condition index and leaf area index. *Computers and Electronics in Agriculture*, 160 , pp. 82-90, 10.1016/j.compag.2019.03.017
48. Weiss, U., Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and Autonomous Systems*, 59, 265-273, 10.1016/j.robot.2011.02.011
49. Wijesingha, J., Moeckel, T., Hensgen, F., Wachendorf, M. (2019). Evaluation of 3D point cloud-based models for the prediction of grassland biomass. *International Journal of Applied Earth Observation and Geoinformation*, 78, 352-359, 10.1016/j.jag.2018.10.006

50. Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.J. (2017). Big Data in Smart Farming – A review. *Agricultural Systems*, 153, pp. 69-80, 10.1016/j.agsy.2017.01.023

51. Zaman, S., Comba, L., Biglia, A., Ricauda Aimonino, D., Barge, P., Gay, P. (2019). Cost-effective visual odometry system for vehicle motion control in agricultural environments. *Computers and Electronics in Agriculture*, 162, 82-94, 10.1016/j.compag.2019.03.037

52. Zeybek, M., Şanlıoğlu, İ. (2019). Point cloud filtering on UAV based point cloud. *Measurement*, 133, pp. 99-111, 10.1016/j.measurement.2018.10.013

4. Cost-effective visual odometry system for vehicle motion control in agricultural environments

Abstract

In precision agriculture, innovative cost-effective technologies and new improved solutions, aimed at making operations and processes more reliable, robust and economically viable, are still needed. In this context, robotics and automation play a crucial role, with particular reference to unmanned vehicles for crop monitoring and site-specific operations. However, unstructured and irregular working environments, such as agricultural scenarios, require specific solutions regarding positioning and motion control of autonomous vehicles.

In this chapter, a reliable and cost-effective monocular visual odometry system, properly calibrated for the localisation and navigation of tracked vehicles on agricultural terrains, is presented. The main contribution of this work is the design and implementation of an enhanced image processing algorithm, based on the cross-correlation approach. It was specifically developed to use a simplified hardware and a low complexity mechanical system, without compromising performance. By providing sub-pixel results, the presented algorithm allows to exploit low resolution images, thus obtaining high accuracy in motion estimation with short computing time. The results, in terms of odometry accuracy and processing time, achieved during the in-field experimentation campaign on several terrains, proved the effectiveness of the proposed method and its fitness for automatic control solutions in precision agriculture applications.

Keywords: Precision agriculture; Visual odometry; Unmanned ground vehicle (UGV); Real- time image processing; Agricultural field robots

Nomenclature

$\mathrm{CEP}_{\varepsilon_s}$	Circular error probable of translation assessment errors [mm]
$d_{\mathrm{i,j}}$	Digital number of pixel located at i$^{\mathrm{th}}$ row and j$^{\mathrm{th}}$ column of image I
$\bar{d}_{\mathrm{u,v}}$	Average values of digital numbers within a portion of image I
$[f_{\mathrm{x}}, f_{\mathrm{y}}]$	x and y component of image focal length [pixel]
g_{x}	Image pixels spatial resolution [mm/pixel]
g_{y}	Image pixels spatial resolution [mm/pixel]
h_{c}	Camera height from the ground [mm]
I_{k}	Acquired grey scale image at time instant t_{k}
$\ell_{\mathrm{i,j}}$	Digital number of pixel located at i$^{\mathrm{th}}$ row and j$^{\mathrm{th}}$ column of image L
$\bar{\ell}$	Average values of digital numbers within template $T(\vartheta)$
$L_{\mathrm{k}}(\vartheta)$	Image obtained by rotating image I_{k} by angle ϑ
n_{Γ}	Distance threshold from γ_{M}
m	Coefficient to set the threshold values for γ
N_{i} x N_{j}	Image size (height x width) [pixel]
$O_{\mathrm{k}}^{\{UGV\}_{\mathrm{k}}}$	Origin of the $\{UGV\}_{\mathrm{k}}$ reference frame at time t_{k}
p_{T}	Template size
$p_{\mathrm{i,j}}^{\{UGV\}_{\mathrm{k}}}$	Position of pixel $d_{\mathrm{i,j}}$ in the reference frame $\{UGV\}_{\mathrm{k}}$ at time t_{k} [mm]
$p_{\hat{u},\hat{v}}^{\{UGV\}_{\mathrm{k+1}}}$	Position of the template $T_{\mathrm{k}}(\hat{\vartheta})$ centre in image $I_{\mathrm{k+1}}$ [mm]
$[p_{\mathrm{c,x}}, p_{\mathrm{c,y}}]^T$	Position coordinates of the camera centre in the $\{UGV\}_{\mathrm{k}}$ reference frame [mm]

$q(u, v, \vartheta)$	Binary function to select a neighbourhood Γ of $\gamma(u, v, \vartheta)$
$R(\cdot)$	Rotation matrix
$s(\cdot)$ (or $s_{\mathrm{k}}^{\mathrm{k+1}}(\cdot)$)	Evaluated vehicle translation (between time instant t_{k} and $t_{\mathrm{k+1}}$) [mm]
s_{r}	Reference vehicle translation [mm]
t_{k}	Generic image acquisition time instant [s]
$T_{\mathrm{k}}(\vartheta)$	Pixel subset, called template, of image $L_{\mathrm{k}}(\vartheta)$
U	Ordered set of u indices
$[\hat{u}_{\mathrm{e}}, \hat{v}_{\mathrm{e}}, \hat{\vartheta}_{\mathrm{e}}]$	Weighted centroid of Γ
$\{UGV\}_{\mathrm{k}}$	Reference frame of the UGV at time t_{k}
V	Ordered set of v indices
w_{T}	Semi-width of the template T_{k} [pixels]
Greek letters	
$\gamma(u, v, \vartheta)$	Normalised cross-correlation function
γ_{M}	Maximum value of $\gamma(u, v, \vartheta)$
δ_{ϑ}	Angular resolution of the VO process [deg]
Γ	Specific subset of γ
ε_{s}	Error in translation assessment between two successive images [mm]
ε_{ϑ}	Error in orientation assessment between two successive images [deg]
ϑ	Rotation angle of image $L_{\mathrm{k}}(\vartheta)$ [deg]
$\hat{\vartheta}$	Evaluated vehicle rotation [deg]
ϑ_{r}	Reference vehicle rotation [deg]
$\vartheta_{\min}$	Minimum value of $\vartheta \in \Theta$ [deg]
$\vartheta_{\max}$	Maximum value of $\vartheta \in \Theta$ [deg]
Θ	Ordered set of all considered rotation angles ϑ ($\Theta = \{\vartheta_{\min}, \vartheta_{\min} + \delta_{\vartheta}, \dots, \vartheta_{\max}\}$) [deg]
$\mu_{\varepsilon_{\vartheta}}$	Average of rotation assessment errors [deg]

σ_{ε_s}	Standard deviation of translation assessment errors [mm]
$\sigma_{\varepsilon_\vartheta}$	Standard deviation of rotation assessment errors [deg]

Acronyms

CCD	Charged coupled device
CEP	Circular error probable
GPS	Global positioning system
GSD	Ground sample distance
IMU	Inertial measurement unit
NCC	Normalised cross correlation
PA	Precision agriculture
SSD	Sum of squared differences
UGV	Unmanned ground vehicle
VO	Visual odometry

4.1. Introduction

Precision agriculture (PA) has been recognised as an essential approach to optimise crop-managing practices and to improve field products quality ensuring, at the same time, environmental safety (Ding et al., 2018; Grella et al., 2017; Lindblom et al., 2017). In very large fields and/or in-fields located on hilly areas, cropland monitoring and maintenance may result in a laborious task, requiring automatic machines and procedures (Comba et al., 2018; Grimstad et al., 2017). In this regard, unmanned ground vehicles (UGVs) are playing a crucial role in increasing efficiency in cultivation, e.g. in optimising the use of fertilisers or precision weed control (Utstumo et al., 2018; Vakilian and Massah, 2017; De Baerdemaeker, 2013).

To perform agricultural in-field tasks with the least amount of human interaction, UGVs should be characterised by a high level of automation (van Henten et al., 2013; Kassler, 2001). Nowadays, developed autonomous navigation systems, which use GPS technologies (Bonadies and Gadsden, 2018) and/or machine vision approaches (García-Santillán et al., 2017), allow UGVs, for example, to follow crop rows autonomously, even in complex agricultural scenarios. A common requirement for these applications is a robust up-to-date position and orientation assessment during movements (Ghaleb et al., 2017). Despite the wide diffusion of GPS systems, they show limitations and drawbacks when high precision navigation is required or where the satellite signal is poor, e.g. in covered areas, greenhouses or peculiar hilly regions (Ericson and Åstrand, 2018; Aboelmagd et al., 2013). In agricultural environments, UGV motion estimation by wheel odometry also encounters critical limitations due to wheels slippage on sloped terrains, which is very typical in some crops such as vineyards (Bechar and Vigneault, 2016; Aboelmagd et al., 2013; Nourani-Vatani et al., 2009).

Visual odometry (VO), the measurement of the position and orientation of a system by exploiting the information provided by a set of successive images (Moravec, 1980), can provide reliable movement feedback in UGV motion control (Aqel et al., 2016; Scaramuzza and Fraundorfer, 2011). The hardware required to implement a VO system consists of one or more digital cameras, an image processing unit and an optional lighting system. Not requiring external signals or references, visual odometry has been proven to be very significant in particular contexts where the GPS signal is weak or absent (even where the magnetic field cannot be exploited by compass), by overcoming the limitations of other methodologies (Scaramuzza and Fraundorfer, 2011).

Two main typologies of VO systems can be defined on the basis of the adopted number of cameras: (1) stereo systems use data provided by multiple cameras while (2) monocular systems, characterised by a simple and cost-effective setup, exploit a single digital camera. The image processing of stereo systems is typically complex and time consuming and requires accurate calibration procedures; indeed, an unsynchronised shutter speed between the stereo cameras can lead to errors in motion estimation (Aqel et al., 2016; Jiang et al., 2014). However, the stereo system degrades to the monocular case when the stereo baseline (the distance between the two cameras) is small compared to the distance of the acquired scene by the cameras (Aqel et al., 2016).

The available image processing algorithms for VO applications have two main approaches: (1) feature-based algorithms and (2) appearance-based algorithms. In feature-based VO, specific features/details detected and tracked in the sequence of successive images are exploited (Fraundorfer and Scaramuzza, 2012). Depending on the application, the performance to be achieved and the different approaches in feature selection, several algorithms can be found in literature, such as Libviso (Geiger et al., 2012), Gantry (Jiang et al., 2014) or the Newton-Raphson search methods (Shi and Tomasi, 1994). A different approach is adopted in appearance based-algorithms where successive image frames are searched for changes in appearance by extracting information regarding pixels displacement. The template matching process, which is a widely recognised approach among VO appearance-based solutions, consists in selecting a small portion within a frame (called template) and in comparing it with a temporally subsequent image, then scoring the quality of the matching (Gonzalez et al., 2012; Goshtasby et al., 1984). This task has mainly been performed by using the sum of squared differences (SSD) and normalised cross-correlation (NCC) as similarity measures (Aqel et al., 2016; Yoo et al., 2014; Nourani-Vatani et al., 2009). This latter matching measure, even if computationally heavier than SSD, is invariant to the linear gradient of image contrast and brightness (Mahmood and Khan, 2012; Lewis, 1995).

Motion assessment by VO systems has been proven to be particularly effective when integrated with other sensors such as the inertial measurement unit (IMU), compass sensor, visual compass (Gonzalez et al., 2012), GPS technology or encoders (e.g. on wheels and tracks), to avoid error accumulation on long missions (Zaidner and Shapiro, 2016). Indeed, with particular attention to agricultural applications, innovative and reliable solutions should be developed to reduce system complexity and costs by implementing smart algorithms and by exploiting data fusion (Comba et al., 2016; Zaidner and Shapiro, 2016).

In this chapter, a reliable and cost-effective monocular visual odometry system, properly calibrated for the localisation and navigation of tracked vehicles on agricultural terrains, is presented. The main contribution of this work is the design and implementation of an enhanced image processing algorithm, based on the cross-correlation approach, with sub-pixel capabilities. It was specifically developed to use a simplified hardware and a low complexity mechanical system, without compromising performance. In the implemented VO system, installed on a full electric tracked UGV, ground images acquisition was performed by an off-the-shelf camera. The performance of the system, in terms of computing time and of movement evaluation accuracy, was investigated with in-field tests on several kinds of terrains, typical of agricultural scenarios. In addition, the optimal set of algorithm parameters was investigated for the specific UGV navigation/motion control for precision agricultural applications.

This chapter is structured as follows: Section 4.2 reports the description of the implemented tracked UGV and of the vision system. The proposed algorithm for visual odometry is presented in Section 4.3, while the results from the in-field tests are discussed in Section 4.4. Section 4.5 reports the conclusion and future developments.

4.2. System setup

The implemented VO system was developed to perform the motion and positioning controls of a full electric UGV specifically designed for precision spraying in tunnel crop management, where GPS technology is hampered by metal enclosures. Image acquisition is performed by a Logitech C922 webcam, properly positioned in the front part of the vehicle, with a downward looking setup at the height (h_c) of 245 mm from the ground. The camera having 3 mega pixels had a max resolution of 1080p/30 fps - 720p/ 60 fps. To improve the quality of the acquired images, the camera was shielded with a properly sized rigid cover to protect the portion of ground within the camera field of view from direct lighting, thus avoiding irregular lighting and the presence of marked shadows. The illumination of the observed ground surface is provided by a lighting system made of 48 SMD LED 5050 modules (surface-mount device light-emitting diode) with an overall lighting power of more than 1,000 lumens and a power consumption of 8.6 W. Fig. 4.1 reports the diagram of the VO system setup together with an image of the implemented UGV system.

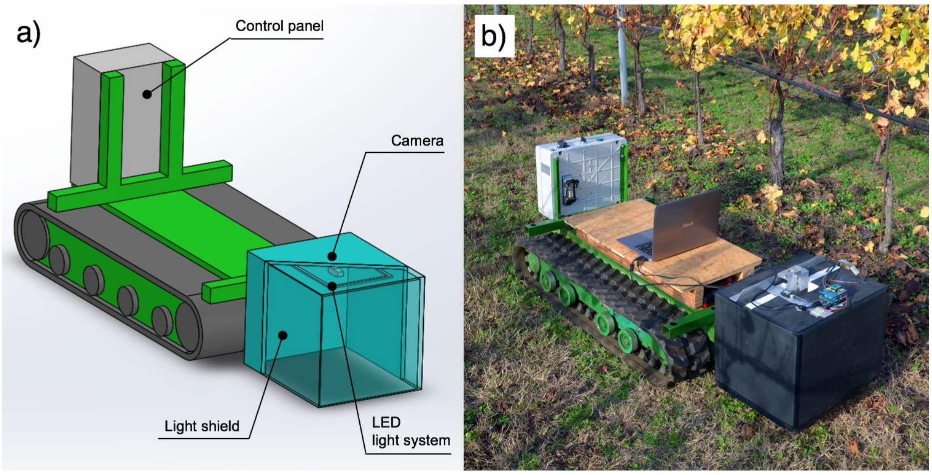


Figure 4.1. Scheme (a) and picture (b) of the implemented UGV prototype. In the final version of the visual odometry system, the lower part of the shielding rigid cover was replaced by a dark curtain.

The image acquisition campaign was conducted on five different terrains (soil, grass, concrete, asphalt and gravel), typical of agricultural environments, in order to assess and quantify the performance of the proposed algorithm. Two datasets of more than 16,000 pairs of grey scale images (8-bit colour representation), at two image resolutions, were processed. Images with a high-resolution have a size of 1280x720 pixels (width and height) while low-resolution ones, which were obtained by down sampling the high resolution ones, are 320x240 pixels (width and height). The sample images at high and low resolution, acquired on five different terrains, are shown in Fig. 4.2.

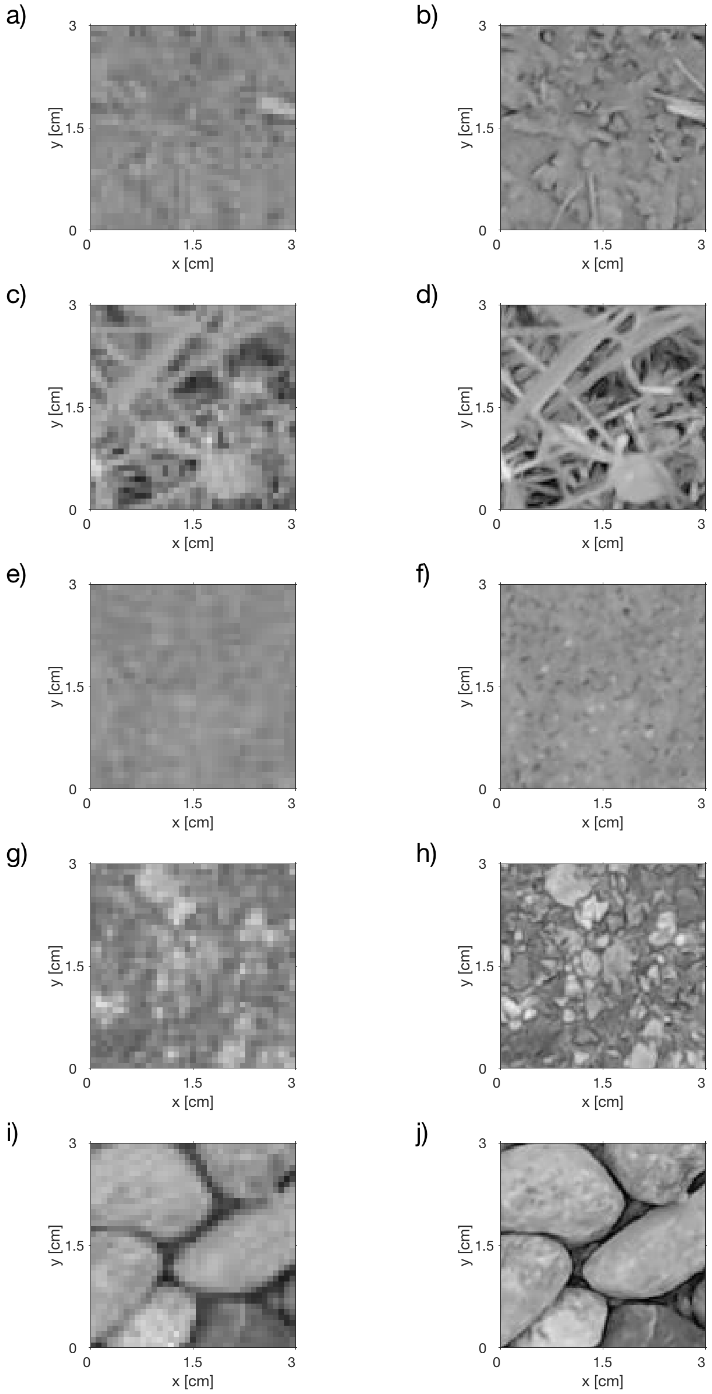


Figure 4.2. Samples of greyscale images of soil (a-b), grass (c-d), concrete (e-f), asphalt (g-h) and gravel (i-j), at low and high resolution, respectively.

A grey scale image I_{k}, acquired at time instant t_{k}, can be defined as an ordered set of digital numbers $d_{\mathrm{i,j}}$ as

$$I_{\mathrm{k}} = \left\{ d_{\mathrm{i,j}} \in [0,1,\dots,255] \vee 1 \leq \mathrm{i} \leq N_{\mathrm{i}}, 1 \leq \mathrm{j} \leq N_{\mathrm{j}} \right\} \tag{1}$$

where i and j are the row and column indices while N_{i} and N_{j} are the numbers of pixels per row and column, respectively.

The intrinsic camera parameters and acquisition settings were evaluated by performing a calibration procedure (Matlab© calibration toolbox). The focal length in pixel was $\left(f_{\mathrm{x}}, f_{\mathrm{y}}\right) = (299.4122, 299.4303)$ and $\left(f_{\mathrm{x}}, f_{\mathrm{y}}\right) = (888.5340, 888.8749)$ for the low-resolution and high-resolution images respectively. The position [mm] of pixels $d_{\mathrm{i,j}}$ in the UGV reference frame $\{UGV\}_{\mathrm{k}}$ at time t_{k}, defined with origin O_{k} in the barycentre of the tracked system and with the x-axis aligned to the vehicle's forward motion direction (Fig. 4.4), can thus be easily computed as

$$p_{\mathrm{i,j}}^{\{UGV\}_{\mathrm{k}}} = \left[\left(\mathrm{j} - \left\lceil \frac{N_{\mathrm{j}}}{2} \right\rceil \right) \frac{h_{\mathrm{c}}}{f_{\mathrm{x}}}, \left(\left\lceil \frac{N_{\mathrm{i}}}{2} \right\rceil - \mathrm{i} \right) \frac{h_{\mathrm{c}}}{f_{\mathrm{y}}} \right]^{\mathrm{T}} + \left[p_{\mathrm{c,x}}, p_{\mathrm{c,y}} \right]^{\mathrm{T}} \tag{2}$$

where $\frac{h_{\mathrm{c}}}{f_{\mathrm{x}}}$ and $\frac{h_{\mathrm{c}}}{f_{\mathrm{y}}}$ are the pixels' spatial resolutions g_{x} and g_{y} [mm/pixel] respectively and $\left[p_{\mathrm{c,x}}, p_{\mathrm{c,y}}\right]^{\mathrm{T}}$ are the position coordinates of the camera centre [mm] in the $\{UGV\}_{\mathrm{k}}$. In the implemented UGV, the position coordinates of the camera with respect to the barycentre of the tracked system are $[950,0]^{\mathrm{T}}$ mm. The relevant camera and images intrinsic parameters adopted in this work are summarised in Table 4.1.

Table 4.1. Intrinsic parameters of the camera and of the processed images

Image type	N_i	N_j	f_x (pixels)	f_y (pixels)	g_x (mm/pixel)	g_y (mm/pixel)
Low-resolution	320	240	299.4303	299.4122	0.8182	0.8183
High-resolution	1280	720	888.8749	888.5340	0.2756	0.2757

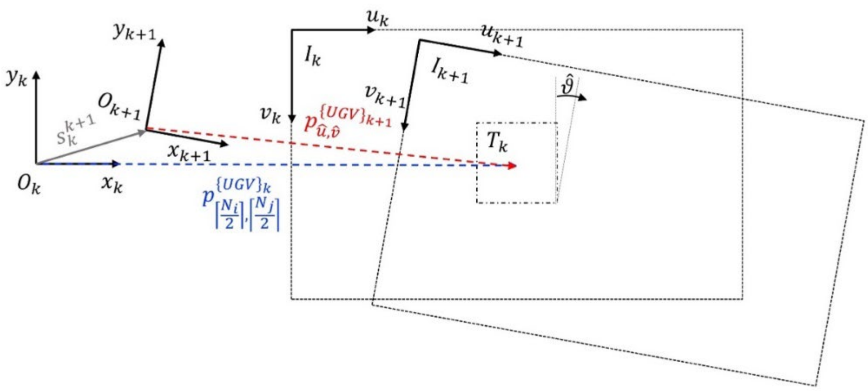


Figure 4.4. Visual odometry variables layout: position $p_{\lceil \frac{N_\mathrm{i}}{2} \rceil, \lceil \frac{N_\mathrm{j}}{2} \rceil}^{\{UGV\}_\mathrm{k}}$ of template T_k in the UGV reference frame $\{UGV\}_\mathrm{k}$ (x_k and y_k axis with O_k origin); position $p_{\hat{u},\hat{\vartheta}}^{\{UGV\}_\mathrm{k+1}}$ of template $T_\mathrm{k}(\hat{\vartheta})$ in the updated UGV reference frame $\{UGV\}_\mathrm{k+1}$ ($x_\mathrm{k+1}$ and $y_\mathrm{k+1}$ axis with $O_\mathrm{k+1}$ origin); rotation angle $\hat{\vartheta}$ of image $I_\mathrm{k+1}$ with respect to I_k and UGV evaluated movement assessment $s_\mathrm{k}^\mathrm{k+1}$.

4.3. Visual odometry algorithms

In visual odometry, the objective of measuring the position and orientation of an object at time $t_{\mathrm{k+1}}$, knowing its position and orientation at time t_{k}, is performed by evaluating the relative movement of a solid camera having occurred during time interval $t_{\mathrm{k+1}} - t_{\mathrm{k}}$. This task is performed by comparing the image pair I_{k} and $I_{\mathrm{k+1}}$, acquired in the ordered time instants t_{k} and $t_{\mathrm{k+1}}$, respectively.

In the normalised cross-correlation (NCC) approach, a pixel subset $T_{\mathrm{k}}(\vartheta)$ (also named template) is selected from the image $L_{\mathrm{k}}(\vartheta)$ centre, which is obtained rotating image I_{k} by an angle ϑ, as

$$T_{\mathrm{k}}(\vartheta) = \left\{\ell_{\mathrm{i,j}} \in L_{\mathrm{k}}(\vartheta) \middle| \left|\mathrm{i} - \left[\frac{N_{\mathrm{i}}}{2}\right]\right| \leq w_T, \left|\mathrm{j} - \left[\frac{N_{\mathrm{j}}}{2}\right]\right| \leq w_T\right\} \tag{3}$$

where $\ell_{\mathrm{i,j}}$ is a digital number of image L_{k} and w_{T} is the semi-width [pixels] of the template T_{k}. The adopted template size p_{T} can be defined as a fraction of the shortest image dimension as $p_{\mathrm{T}} = 2 \cdot w_{\mathrm{T}} \cdot {N_{\mathrm{i}}}^{-1}$; with this definition $p_{\mathrm{T}} \subset [0\ 1]$. With no assumption on the performed movement, angle ϑ is usually selected from an ordered set of values $\Theta = \{\vartheta_{\min}, \vartheta_{\min} + \delta_{\vartheta}, \dots, \vartheta_{\max}\}$, with $\vartheta_{\min}$ and $\vartheta_{\max}$ chosen to consider the whole circle angle. The δ_{ϑ} parameter can be defined as the angular resolution of the process.

The relative movement of $I_{\mathrm{k+1}}$ with respect to image I_{k}, in terms of translation $[\hat{u}, \hat{v}]^{\mathrm{T}}$ [pixels] and rotation $\hat{\vartheta}$ [deg], is thus performed by assessing the position of the ground portions represented in templates $T_{\mathrm{k}}(\vartheta)$ in the subsequent image $I_{\mathrm{k+1}}$ by solving the problem

$$\gamma_{\mathrm{M}} = \max_{\hat{u},\hat{v},\hat{\vartheta}} \gamma(u, v, \vartheta) \tag{4}$$

with $u \in U = \{w_\mathrm{T}, w_\mathrm{T}+1, \dots, N_\mathrm{i} - w_\mathrm{T}\}$, $v \in V = \{w_\mathrm{T}, w_\mathrm{T}+1, \dots, N_\mathrm{j} - w_\mathrm{T}\}$, $\vartheta \in \Theta$ and where $\gamma(u, v, \vartheta)$ is the normalised cross-correlation function (Aqel et al., 2016; Lewis, 1995) defined as

$$\gamma(u,v,\vartheta) = \frac{\sum_{\mathrm{i}=-w_\mathrm{T}}^{w_\mathrm{T}} \sum_{\mathrm{j}=-w_\mathrm{T}}^{w_\mathrm{T}} \left(d_{\mathrm{i}+u,\mathrm{j}+v} - \bar{d}_{u,v}\right)_{I_{\mathrm{k}+1}} \cdot \left(\ell_{\mathrm{i}+w_\mathrm{T},\mathrm{j}+w_\mathrm{T}} - \bar{\ell}\right)_{T_\mathrm{k}(\vartheta)}}{\sqrt{\sum_{\mathrm{i}=-w_\mathrm{T}}^{w_\mathrm{T}} \sum_{\mathrm{j}=-w_\mathrm{T}}^{w_\mathrm{T}} \left(d_{\mathrm{i}+u,\mathrm{j}+v} - \bar{d}\right)_{I_{\mathrm{k}+1}}^2 \cdot \left(\ell_{\mathrm{i}+w_\mathrm{T},\mathrm{j}+w_\mathrm{T}} - \bar{\ell}\right)_{T_\mathrm{k}(\vartheta)}^2}} \qquad (5)$$

with

$$\bar{d}_{u,v} = \frac{\sum_{\mathrm{i}=-w_\mathrm{T}}^{w_\mathrm{T}} \sum_{\mathrm{j}=-w_\mathrm{T}}^{w_\mathrm{T}} \left(d_{\mathrm{i}+u,\mathrm{j}+v}\right)_{I_{\mathrm{k}+1}}}{4 \cdot w_\mathrm{T}^2} \qquad (6)$$

and

$$\bar{\ell} = \frac{\sum_{\mathrm{i}=-w_\mathrm{T}}^{w_\mathrm{T}} \sum_{\mathrm{j}=-w_\mathrm{T}}^{w_\mathrm{T}} \left(\ell_{\mathrm{i}+w_T,\mathrm{j}+w_T}\right)_{T_\mathrm{k}(\vartheta)}}{4 \cdot w_\mathrm{T}^2} \qquad (7)$$

the average values of the digital numbers within a portion of image $I_{\mathrm{k}+1}$ and template $T_\mathrm{k}(\vartheta)$, respectively. A scheme of the implemented NCC algorithm is reported in Fig. 4.3.

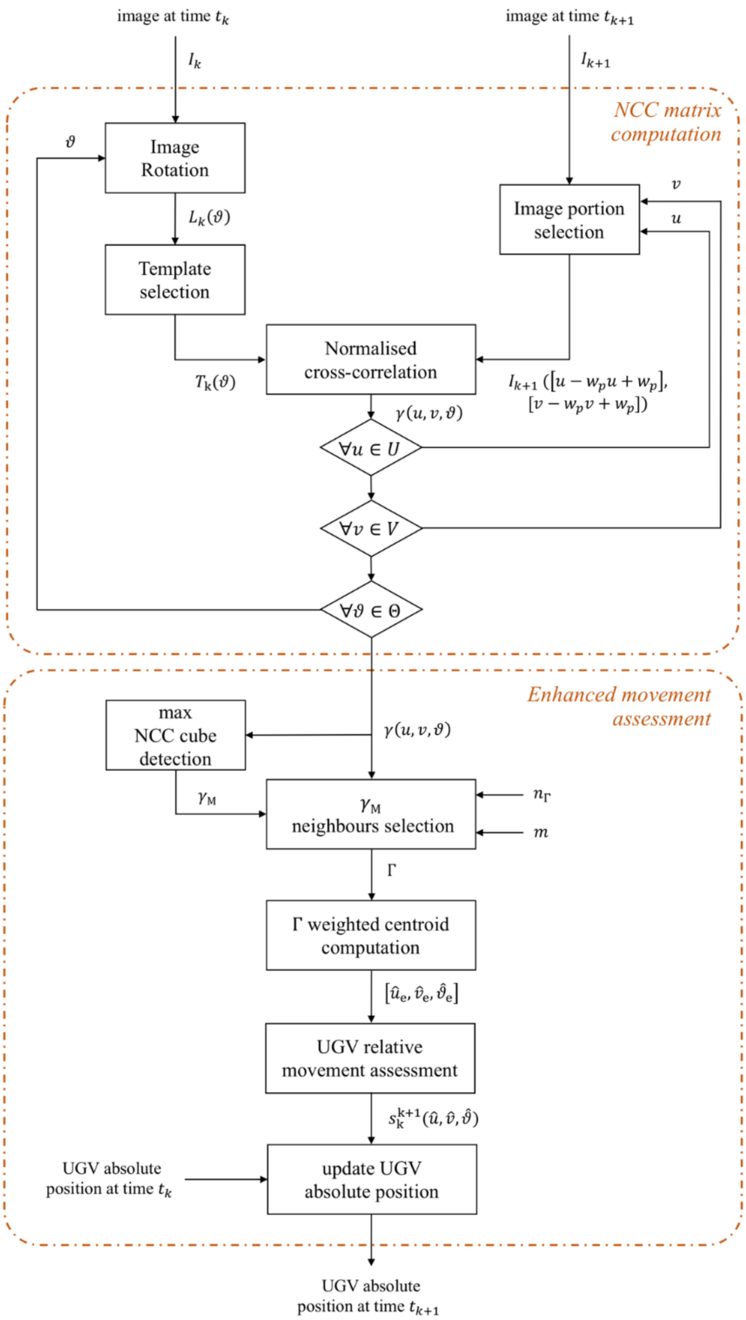


Figure 4.3. Scheme diagram of the implemented enhanced VO algorithm

2098

The relative movement $s_{\mathrm{k}}^{\mathrm{k+1}}$ performed by the UGV in the time interval $t_{\mathrm{k+1}} - t_{\mathrm{k}}$ (Fig. 4.4) can thus be easily computed as

$$s_{\mathrm{k}}^{\mathrm{k+1}}(\hat{u}, \hat{v}, \hat{\vartheta}) = R(-\hat{\vartheta}) \cdot p_{\hat{u},\hat{v}}^{\{UGV\}_{\mathrm{k+1}}} - p_{\left\lceil\frac{N_{\mathrm{i}}}{2}\right\rceil,\left\lceil\frac{N_{\mathrm{j}}}{2}\right\rceil}^{\{UGV\}_{\mathrm{k}}} \tag{8}$$

where $R(-\hat{\vartheta})$ is the rotation matrix of angle $-\hat{\vartheta}$, $p_{\hat{u},\hat{v}}^{\{UGV\}_{\mathrm{k+1}}}$ is the template $T_{\mathrm{k}}(\hat{\vartheta})$ assessed position [mm] in $I_{\mathrm{k+1}}$ (represented in $\{UGV\}_{\mathrm{k+1}}$, Eq. (2)), and $p_{\left\lceil\frac{N_{\mathrm{i}}}{2}\right\rceil,\left\lceil\frac{N_{\mathrm{j}}}{2}\right\rceil}^{\{UGV\}_{\mathrm{k}}}$ is the known position [mm] of template T_{k} in I_{k}, (represented in $\{UGV\}_{\mathrm{k}}$, Eq. (2) . For the sake of clarity, it should be noted that $p_{\left\lceil\frac{N_{\mathrm{i}}}{2}\right\rceil,\left\lceil\frac{N_{\mathrm{j}}}{2}\right\rceil}^{\{UGV\}_{\mathrm{k}}}$ is equal to $\left[p_{\mathrm{c,x}}, p_{\mathrm{c,y}}\right]^{\mathrm{T}}$, which is $[950{,}0]^{\mathrm{T}}$ millimetres, and that $s_{\mathrm{k}}^{\mathrm{k+1}}(\hat{u}, \hat{v}, \hat{\vartheta})$ coincides with $O_{\mathrm{k+1}}^{\{UGV\}_{\mathrm{k}}}$, which is the origin of the reference frame $\{UGV\}_{\mathrm{k+1}}$ represented in $\{UGV\}_{\mathrm{k}}$ reference frame (Fig. 4.4).

4.3.1 Enhanced cross-correlation algorithm

The quality of the UGV's movement measure, using normalised cross-correlation-based visual odometry algorithms, is strictly related to the solution of the problem defined in Eq. (4). The approach of considering the sole maximum value γ_{M} of $\gamma(u, v, \vartheta)$, with $u \in \{w_{\mathrm{T}}, w_{\mathrm{T}} + 1, \dots, N_{\mathrm{i}} - w_{\mathrm{T}}\}$, $v \in \{w_{\mathrm{T}}, w_{\mathrm{T}} + 1, \dots, N_{\mathrm{j}} - w_{\mathrm{T}}\}$ and $\vartheta \in \Theta$, has intrinsic limitations regarding maximum achievable accuracy. Indeed, the digital discretisation of the field of view performed by the digital camera and the discrete set Θ of the investigated orientation ϑ affect both the translation and the rotation assessments. The accuracy of the VO system is thus related to the adopted image resolution, being directly related to the pixels ground sample distance (GSD) g_{x} and g_{y} and the angle step δ_{ϑ}

adopted in the image processing. Regarding this aspect, an accuracy improvement can be pursued by adopting high-resolution cameras, which can provide images with smaller pixels GSD g_{x} and g_{y}: favourable effects are linked, in the meanwhile, to the accuracy of $[\hat{u}, \hat{v}]^{\mathrm{T}}$ and to the angular resolution δ_{ϑ} values. Indeed, concerning the rotation procedure of image $L_{\mathrm{k}}(\delta_{\vartheta})$, if the rotation angle δ_{ϑ} is small, no modifications are obtained on the pixels' digital number in the central part of the image, where the template is selected. For the sake of clarity, the smallest δ_{ϑ} values which lead to template $T_{\mathrm{k}}(\delta_{\vartheta})$ modifications, in relation to image resolution and template size p_{T}, are reported in Table 4.2.

Table 4.2. Angular resolution $\delta_{\vartheta,\min}$ as a function of the template size p_{T}

		Template size $\boldsymbol{p}_{\mathbf{T}}$		
		0.1	0.2	0.3
Image resolution	320x240	2.24 [deg]	1.15 [deg]	0.77 [deg]
	1280x720	0.77 [deg]	0.39 [deg]	0.26 [deg]

However, increasing image resolution leads to a considerable increment in the required computing load, which does not fit with the real-time requirements of the VO algorithm application or requires technologies which are too expensive.

The proposed approach is aimed at increasing VO assessment accuracy by using very low-resolution images, which allows to drastically reduce the computing load while achieving results comparable to the ones obtained by processing high-resolution data. This translates into more cost-effective systems, requiring economical acquisition and processing hardware.

For this purpose, a function $q(u,v,\vartheta)$ was defined as

$$q(u,v,\vartheta) = \begin{cases} 0 & \text{if} \quad \gamma(u,v,\vartheta) < m \cdot \gamma_{\mathrm{M}}, \quad \left\| \left([u,v,\vartheta] - [\hat{u},\hat{v},\hat{\vartheta}] \right) \circ [1,1,\delta_{\vartheta}^{-1}] \right\|_2 > n_{\Gamma} \\ 1 & \text{if} \quad \gamma(u,v,\vartheta) \geq m \cdot \gamma_{\mathrm{M}}, \quad \left\| \left([u,v,\vartheta] - [\hat{u},\hat{v},\hat{\vartheta}] \right) \circ [1,1,\delta_{\vartheta}^{-1}] \right\|_2 \leq n_{\Gamma} \end{cases} \quad (9)$$

in order to consider a neighbourhood Γ of the maximum γ_{M} (Eq. (4)) of cross-correlation discrete function $\gamma(u,v,\vartheta)$ in the space (u,v,ϑ), with values higher than $m \cdot \gamma_{\mathrm{M}}$. In particular, n_{Γ} is the distance threshold from γ_{M} and m is the coefficient to set the γ values threshold. In this work, adopted values are $n_{\Gamma} = 5$ and $m = 0.95$ on the base of empirical evaluations. The Hadamard product with $[1,1,\delta_{\vartheta}^{-1}]$ was adopted to normalise the weight of the three spatial coordinates (u,v,ϑ).

The enhanced movement assessment is thus performed by computing the weighted centroids $[\hat{u}_{\mathrm{e}}, \hat{v}_{\mathrm{e}}, \hat{\vartheta}_{\mathrm{e}}]$ of Γ (Fig. 4.5), as

$$\hat{u}_{\mathrm{e}} = \frac{\sum_{u=w_{\mathrm{T}}}^{N_{\mathrm{i}}-w_{\mathrm{T}}} u \cdot \sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} \sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} \gamma(u,v,\vartheta_{\mathrm{z}}) \cdot q(u,v,\vartheta_{\mathrm{z}})}{\sum_{u=w_{\mathrm{T}}}^{N_{\mathrm{i}}-w_{\mathrm{T}}} \sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} \sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} q(u,v,\vartheta_{\mathrm{z}})} \quad (10)$$

$$\hat{v}_{\mathrm{e}} = \frac{\sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} v \cdot \sum_{u=1}^{N_{\mathrm{i}}-w_{\mathrm{T}}} \sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} \gamma(u,v,\vartheta_{\mathrm{z}}) \cdot q(u,v,\vartheta_{\mathrm{z}})}{\sum_{u=w_{\mathrm{T}}}^{N_{\mathrm{i}}-w_{\mathrm{T}}} \sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} \sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} q(u,v,\vartheta_{\mathrm{z}})} \quad (11)$$

and

$$\hat{\vartheta}_{\mathrm{e}} = \frac{\sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} \mathrm{z} \cdot \sum_{u=w_{\mathrm{T}}}^{N_{\mathrm{i}}-w_{\mathrm{T}}} \sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} \gamma(u,v,\vartheta_{\mathrm{z}}) \cdot q(u,v,\vartheta_{\mathrm{z}})}{\sum_{u=w_{\mathrm{T}}}^{N_{\mathrm{i}}-w_{\mathrm{T}}} \sum_{v=w_{\mathrm{T}}}^{N_{\mathrm{j}}-w_{\mathrm{T}}} \sum_{\mathrm{z}=1}^{\mathrm{card}(\Theta)} q(u,v,\vartheta_{\mathrm{z}})} \quad (12)$$

With the proposed approach, the UGV's movement evaluation is not defined by discrete values, since $[\hat{u}_{\mathrm{e}}, \hat{v}_{\mathrm{e}}, \hat{\vartheta}_{\mathrm{e}}] \in \mathbb{R}^3$.

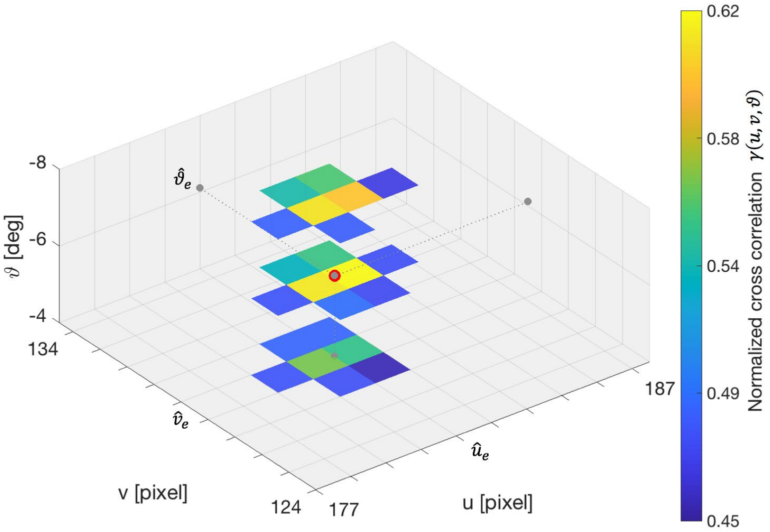


Figure 4.5. 3D cross-correlation matrix $\gamma(u, v, \vartheta)$ and the position coordinates $[\hat{u}_{\mathrm{e}}, \hat{v}_{\mathrm{e}}, \hat{\vartheta}_{\mathrm{e}}]$ of the weighted centroids obtained by the enhanced VO algorithm.

4.4. Results and discussion

The performance of the proposed visual odometry system, developed for a UGV motion estimation, was assessed by processing more than 16,000 images. The in-field tests were performed on different agricultural terrains by acquiring images on soil, grass, asphalt, concrete and gravel. In particular, both rectilinear and curvilinear paths were planned. Considering the whole dataset, the travelled distance between two subsequent images ranges between 0 mm (static vehicle) and 70 mm, which guarantees a minimum overlapping area of 72%. The relative rotation does not exceed the range of [-9 +9] degrees, due to the short movement between two acquired frames. The image resolutions were 1280x720 pixels (high-resolution images) and 320x240 pixels (low-resolution images). To evaluate the performance improvements of the proposed algorithm, with sub-pixel capabilities, the set of acquired images was also processed by means of a standard VO algorithm (Computer Vision System Toolbox, MathWorks, 2018).

The performance analysis of the proposed VO system was performed: (1) by assessing motion evaluation accuracy in pairs of successive images, using high-resolution datasets as a reference, and (2) by computing the cumulative error with respect to in-field position references travelling about 10 meters long paths.

Concerning a pair of successive images, the error in measuring the relative movement s and the rotation ϑ between two subsequent images was defined as

$$\varepsilon_{\mathrm{s}} = \|s(\cdot) - s_{\mathrm{r}}\|_2 \qquad (13)$$

and

$$\varepsilon_{\vartheta} = \left|\hat{\vartheta} - \vartheta_{\mathrm{r}}\right| \qquad (14)$$

respectively, where $s(\cdot)$ (Eq. (8)) and $\hat{\vartheta}$ are the vehicle's movement and rotation, evaluated by using the enhanced and standard algorithm and by processing low-resolution images, while s_{r} and ϑ_{r} represent the reference measurements from the high-resolution images. Concerning the translation assessment, accuracy was expressed by the circular error probable ($\mathrm{CEP}_{\varepsilon_{\mathrm{s}}}$) and standard deviation ($\sigma_{\varepsilon_{\mathrm{s}}}$) indices (Winkler et al., 2012) (Table 4.3), while accuracy in measuring the changes in vehicle orientation ϑ were described by computing the average ($\mu_{\varepsilon_{\vartheta}}$) and standard deviation ($\sigma_{\varepsilon_{\vartheta}}$) of the computed ε_{ϑ} errors (Table 4.4).

Table 4.3. Accuracy in translation evaluation provided by standard and the enhanced algorithms, detailed for different terrains and considering the overall acquired data. Adopted template size $p_{\mathrm{T}} = 0.2$. Achieved percent improvement of the enhanced algorithm is also reported for every evaluation.

Terrains	Standard algorithm accuracy		Enhanced algorithm accuracy		Accuracy improvement	
	$\mathrm{CEP}_{\varepsilon_{\mathrm{s}}}$ [mm]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [mm]	$CEP_{\varepsilon_{\mathrm{s}}}$ [mm]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [mm]	$CEP_{\varepsilon_{\mathrm{s}}}$ [%]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [%]
Soil	0.45	0.19	0.19	0.1	58.48	50.11
Grass	0.35	0.19	0.19	0.14	44.1	24.43
Concrete	0.28	0.14	0.14	0.08	50.64	44.55
Asphalt	0.37	0.11	0.13	0.07	63.31	36.93
Gravel	0.39	0.14	0.16	0.07	57.4	52.29
Overall	0.37	0.16	0.16	0.09	54.79	41.66

Table 4.4. Accuracy in orientation evaluation provided by standard and the enhanced algorithms, detailed for different terrains and considering the overall acquired data. Adopted template size $p_{\mathrm{T}} = 0.2$. Achieved percent improvement of the enhanced algorithm is also reported for every evaluation.

Terrains	Standard Algorithm accuracy		Enhanced algorithm accuracy		Accuracy improvement	
	$\mu_{\varepsilon_\vartheta}$ [deg]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [deg]	$\mu_{\varepsilon_\vartheta}$ [deg]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [deg]	$\mu_{\varepsilon_\vartheta}$ [%]	$\sigma_{\varepsilon_{\mathrm{s}}}$ [%]
Soil	0.75	0.38	0.29	0.24	61.12	37.19
Grass	0.65	0.36	0.42	0.26	34.45	29.23
Concrete	0.96	1.19	0.18	0.14	81.59	88.48
Asphalt	1.08	1.5	0.15	0.15	86.46	90.09
Gravel	0.97	0.87	0.25	0.2	74.3	76.99
Overall	0.88	0.86	0.26	0.2	67.58	64.39

The results were detailed for each in-field test performed on a specific kind of terrain and, finally, computed by considering the whole image dataset. Overall accuracy in the translation assessment of the proposed algorithm across different terrains resulted to be $CEP_{\varepsilon_{\mathrm{s}}} = 0.16$ mm, with an improvement of around 54% with respect to the values obtained by processing the images with the standard algorithm, which shows a $CEP_{\varepsilon_{\mathrm{s}}}$ of 0.37 mm. The average error in the vehicle's orientation assessment was $\mu_{\varepsilon_\vartheta} = 0.26$ degrees, with an improvement of around 67.6% with respect to the values obtained by processing the images with the standard algorithm. The typology of terrain slightly affects the achieved performance: on the grass surface, a lower performance improvement was found compared to other terrains. Indeed, the greater variability in object height within the camera field of view can lead to additional perspective errors. Nevertheless, even in these complex scenarios, improvements of 44% in the CEP_{ε_s} and of 34% in the orientation assessment was observed (CEP_{ε_s}= 0.19 mm and $\mu_{\varepsilon_\vartheta} = 0.42$ degree) compared to the ones obtained by the standard algorithm. Boxplots of errors ε_s and ε_ϑ, computed by considering the whole image dataset, are reported in Fig. 4.6 for standard and enhanced algorithms. The x and y components of ε_s and the CEP_{ε_s} circles are detailed in Fig. 4.7, with ε_ϑ represented by using a colour bar.

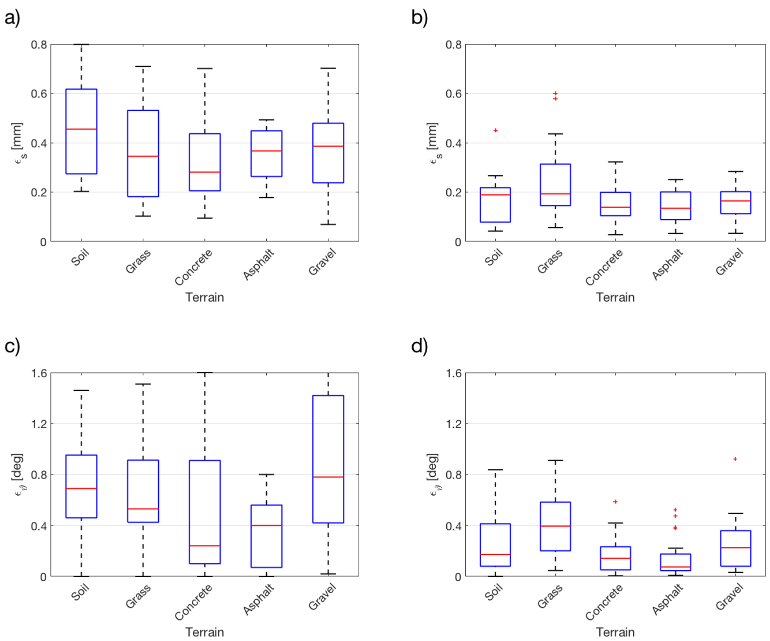


Figure 4.6. Boxplots of translation errors ε_s obtained by standard (a) and enhanced algorithm (b) and of rotation errors ε_ϑ, for standard (c) and enhanced algorithm (d).

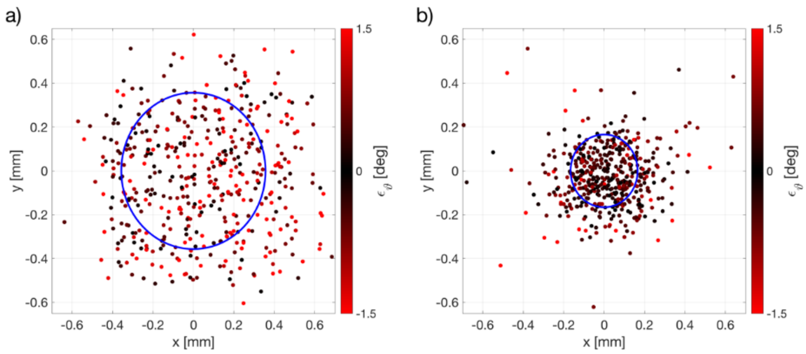


Figure 4.7. Representation of x and y component of errors ε_s obtained by the standard (a) and the enhanced algorithm (b). Errors ε_ϑ are represented with a colormap from black to red. Circle areas bounded by the CEP_{ε_s} is represented with blue solid line.

The cumulative error was computed for 20 sample paths of the tracked vehicle with a length of 9.6 meters, defined as a curvilinear path generated by a sinusoidal trajectory of 0.15 m amplitude and of 3.2 m period. The number of acquired images for a path repetition ranges between 156 and 166, with an average travelled distance between two consecutive frames of 61 mm. Defining a normalised cumulative error with respect to the travelled distance, the obtained values are 0.08 and 0.84 $[\mathrm{deg} \cdot \mathrm{m}^{-1}]$ for what concerns translation and orientation, respectively. The improvement compared to the standard algorithm is of about 60% for both the translation and orientation assessments. The boxplots of all the obtained cumulative errors, expressed in normalised values, are reported in Fig. 4.8.

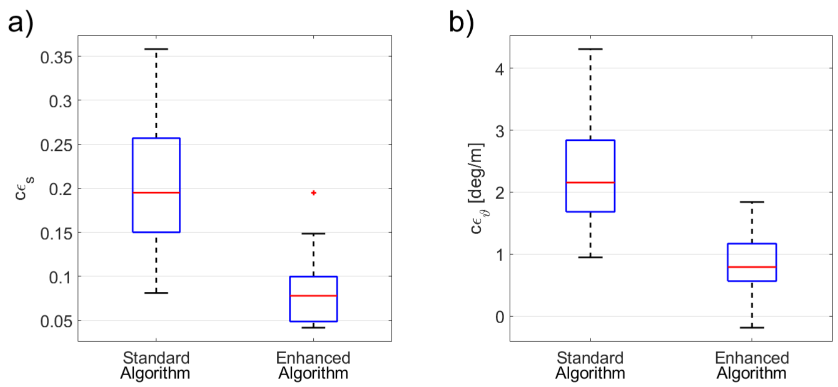


Figure 4.8. Boxplots of normalised cumulative errors of translation $c\varepsilon_s$ (a) and rotation $c\varepsilon_\vartheta$ (b) assessment measured on 20 repetition of 9.6 meters long sample path on several terrains, obtained by standard and enhanced algorithm.

Considering a constant travelled distance, the cumulative error is strictly related to the number of processed images, as every processing step contributes to the overall error. With this assumption, to minimise the cumulative error, pairs of frames acquired at the largest distance, still guaranteeing the proper overlapping surface, should be used. For this purpose, a multi-frame approach can further improve system performance (Jiang et al., 2014).

The optimal configuration for a VO system setup requires thorough analysis of the parameters related to image processing and their tuning according to the application requirements. With particular attention to the overall VO system performance, the size p_{T} of the template $T_{\mathrm{k}}(\vartheta)$ is a relevant algorithm parameter since it is strictly related to (1) the motion accuracy measure, (2) the allowed maximum length of the relative movement between two subsequent images, which should still assure the required overlapping surface of the template, (3) the computing time and, thus, (4) the maximum allowed velocity with a specific VO setup.

The template size p_{T} has a non-linear and non-monotonic effect on the overall VO system's accuracy. Considering the translation assessment, by varying p_{T} within the range 0.05-0.35, an optimal value can be found that provides the best accuracy. Indeed, the proposed algorithm achieves a $CEP_{\varepsilon_{\mathrm{s}}} = 0.16$ mm for $p_{\mathrm{T}} = 0.20$, while accuracy degrades to $CEP_{\varepsilon_{\mathrm{s}}} = 0.21$ mm and $CEP_{\varepsilon_{\mathrm{s}}} = 0.22$ for $p_{\mathrm{T}} = 0.05$ and $p_{\mathrm{T}} = 0.35$, respectively. The boxplots of errors ε_{s} and ε_{ϑ}, obtained by setting p_{T} within the range 0.05-0.35, are reported in Figs. 4.9 and 4.10, respectively.

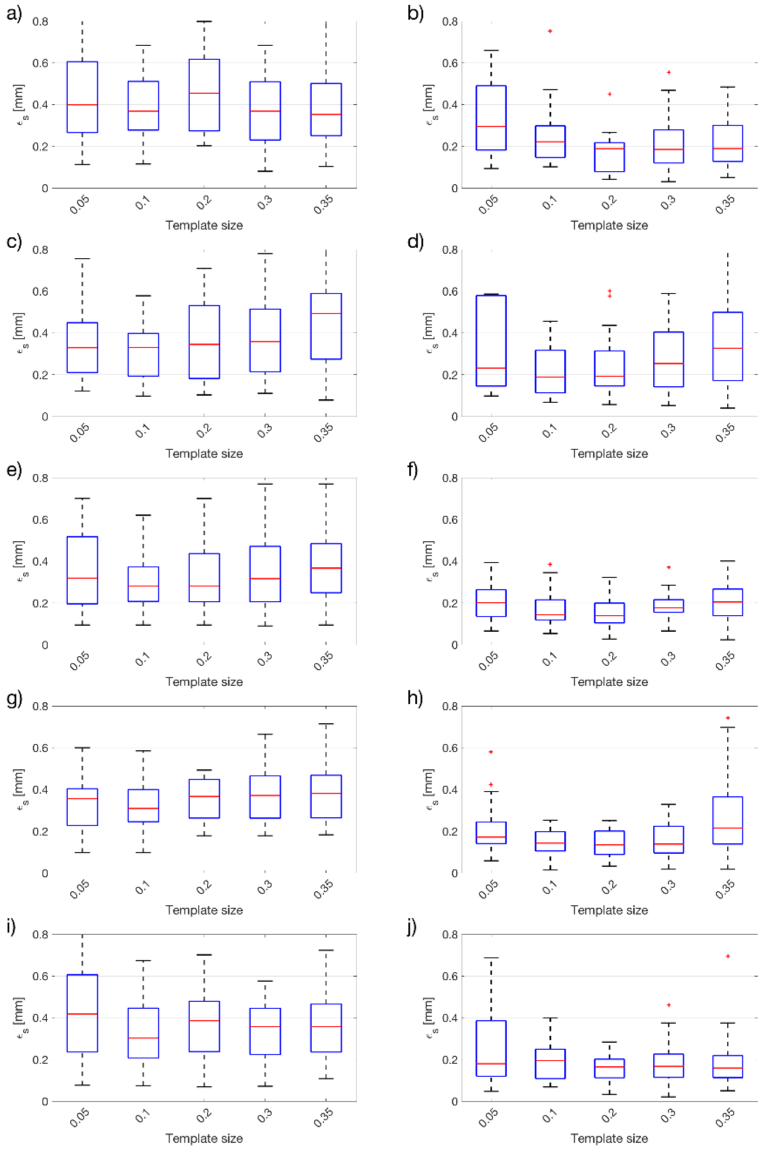


Figure 4.9. Boxplots of accuracy in translation measurement $s(\cdot)$, detailed for algorithm template size p_{T} from 0.05 to 0.4 and for typology of travelled terrain. (soil (a-b), grass (c-d), concrete (e-f), asphalt (g-h) and gravel (i-j))

2279

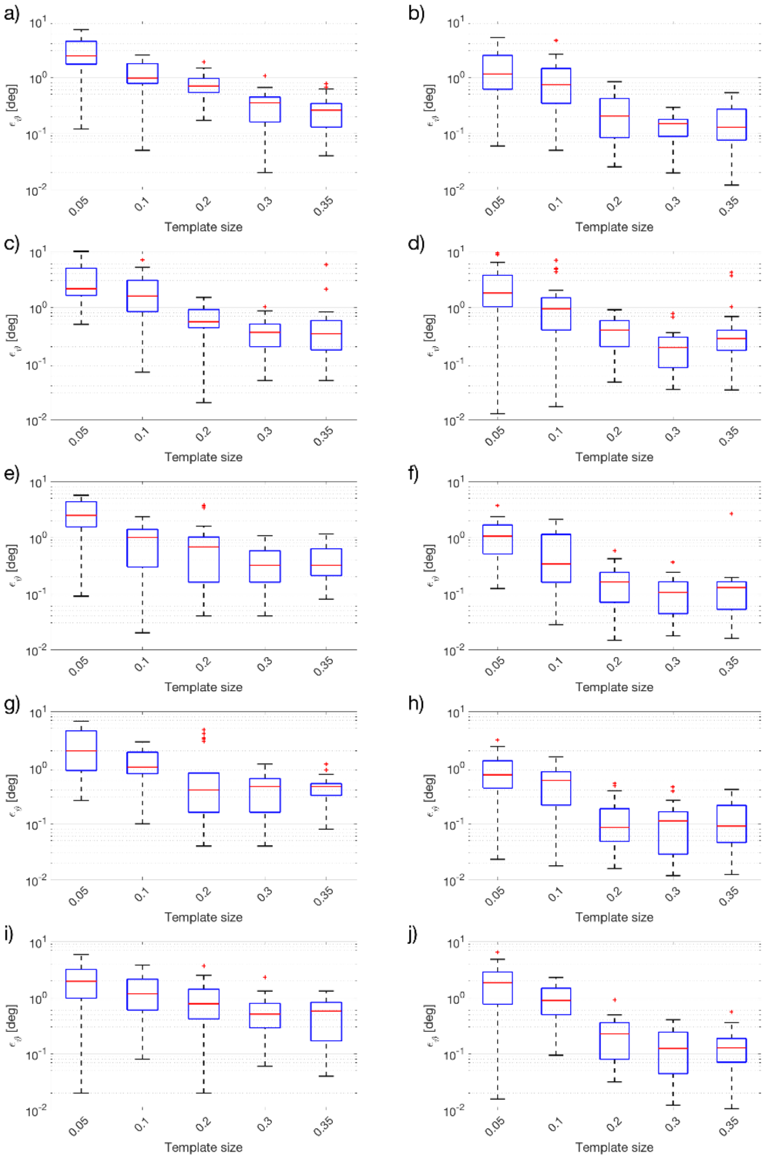


Figure 4.10. Boxplots of accuracy in rotation measurement $\hat{\vartheta}$, detailed for algorithm template size p_{T} from 0.05 to 0.4 and for typology of travelled terrain. (soil (a-b), grass (c-d), concrete (e-f), asphalt (g-h) and gravel (i-j))

The observed accuracy trend in determining the vehicle's orientation is similar to the one described for translation, with the exception of the effect of p_{T} values greater than 0.20 on the accuracy's decrement: it is less marked until p_{T} exceeds 0.6, values that lead to insufficient overlapping surfaces between two successive images. Indeed, regarding proper overlapping surfaces between successive images, the template size should not exceed a certain value. Larger template sizes p_{T} require a shorter relative movement of the vehicle between image acquisition time instants to avoid complete mismatch between a pair of successive images. In the implemented VO system performance evaluation, increasing p_{T} from 0.1 to 0.6 will limit the maximum allowed movement from 93.1 to 39.2 mm, requiring a higher framerate to keep proper image acquisition when considering a constant vehicle velocity.

Concerning the computing time, smaller p_{T} values allow to drastically reduce the required time to process an image pair: considering a low-resolution dataset, the average computing time (0.02 seconds) using $p_{\mathrm{T}} = 0.05$ is 88% shorter than the one required by $p_{\mathrm{T}} = 0.35$ (0.19 seconds). Fig. 4.11a reports the average computing time obtained for processing low and high resolution images with a template size p_{T} ranging from 0.05 to 0.8.

Consequently, the allowed maximum velocity of the vehicle is thus strictly related to template size: considering a constant computing power, smaller template sizes lead to higher vehicle maximum speeds, due to the concurrent effects on the processing time required for an image pair and the length of the maximum allowed movement between two subsequent images. In the implemented VO system, processing low-resolution images by using a value of $p_{\mathrm{T}} = 0.05$, the upper limit velocity (about $4.1\ \mathrm{m \cdot s^{-1}}$) is 91% greater than the one allowed by $p_{\mathrm{T}} = 0.35$ (about

$0.3\ \mathrm{m \cdot s^{-1}}$). The maximum allowed velocities for low and high-resolution images with respect to template size p_{T} ranging from 0.05 to 0.8 are represented in Fig. 4.11b.

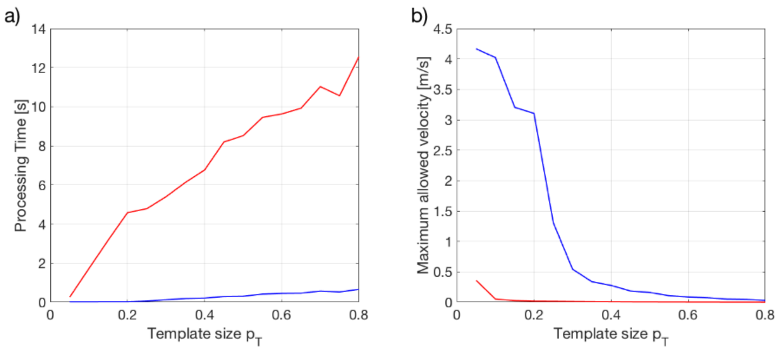


Figure 4.11. Template size p_{T} influence on processing time (a) and maximum allowed UGV velocity (b). Results obtained by processing low-resolution and high-resolution images are represented by blue and red lines, respectively.

4.5. Conclusions

In this chapter, an enhanced image processing algorithm for a cost-effective monocular visual odometry system, aimed at obtaining highly reliable results at low computational costs for a tracked UGV navigation in agricultural applications, is presented. The implemented VO system consists of a downward looking low cost web-camera sheltered with a rigid cover to acquire images with uniform LED lighting. Based on the normalised cross-correlation methodology, the proposed VO algorithm was developed to exploit low-resolution images (320x240 pixels), achieving sub-pixel accuracy in motion estimation. The algorithm allows the VO system to be applied to real-time applications using cost-effective hardware, by requiring a lower computational load.

The robustness of the proposed VO algorithm was evaluated by performing an extensive in-field test campaign on several terrains typical of agricultural scenarios: soil, grass, concrete, asphalt and gravel. The relationship between system performances and more relevant algorithm parameters was investigated in order to determine a proper final system setup.

The obtained overall accuracy, in terms of circular probable error and normalised cumulative error, which are 0.16 mm and 0.08 respectively, were compatible with UGV requirements for precision agricultural applications. The obtained short computing time allowed the vehicle to achieve a maximum velocity limit higher than $4\ \mathrm{m} \cdot \mathrm{s}^{-1}$.

Based on the relative motion assessment, the performance of VO systems degrades when incrementing path length. Therefore, the system integration with absolute reference is required to maintain the needed accuracy during long mission paths.

Acknowledgments

The presented research was partially funded by the Italian Ministry of University and Research PRIN 2015 "Ottimizzazione di macchine operatrici attraverso l’analisi del profilo di missione per un’agricoltura più efficiente” (Prot. 2015KTY5NW) and the “ElectroAgri” Project, Manunet ERA-NET, Call 2016, P.O.R. FESR 2014-2020 European Program.

4.6. References

1. N. Aboelmagd, T.B. Karmat, J Georgy. Fundamentals of inertial navigation, satellite-based positioning and their integration. Springer, (2013), doi:10.1007/978-3-642-30466-8
2. M.O.A. Aqel, M.H. Marhaban, M.I. Saripan, N.Bt. Ismail. Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus, 5 (2016), doi:10.1186/s40064-016-3573-7
3. J. De Baerdemaeker. Precision Agriculture Technology and Robotics for Good Agricultural Practices. IFAC Proceedings Volumes, 46 (2013), pp. 1-4, doi:10.3182/20130327-3-JP-3017.00003
4. Bechar, C. Vigneault. Agricultural robots for field operations: Concepts and components. Biosyst Eng, 149 (2016), pp. 94-111, doi:10.1016/j.biosystemseng.2016.06.014
5. S. Bonadies, S.A. Gadsden. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. Engineering in Agriculture, Environment and Food, 12 (2019), pp. 24-31, doi:10.1016/j.eaef.2018.09.001
6. L. Comba, A. Biglia, D. Ricauda Aimonino, P. Gay. Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. Comput Electron Agr, 155 (2018), pp. 84-95, doi:10.1016/j.compag.2018.10.005
7. L. Comba, P. Gay, D. Ricauda Aimonino. Robot ensembles for grafting herbaceous crops. Biosyst Eng, 146 (2016), pp. 227-239, doi:10.1016/j.biosystemseng.2016.02.012
8. Y. Ding, L. Wang, Y. Li, D. Li. Model predictive control and its application in agriculture: A review. Comput Electron Agr, 151 (2018), pp. 104-117, doi:10.1016/j.compag.2018.06.004
9. S.K. Ericson, B.S. Åstrand. Analysis of two visual odometry systems for use in an agricultural field environment. Biosyst Eng, 166 (2018), pp. 116-125, doi:10.1016/j.biosystemseng.2017.11.009

10. F. Fraundorfer, D. Scaramuzza. Visual odometry: Part II: Matching, robustness, optimization, and applications. IEEE Robotics and Automation Magazine, 19 (2012), pp. 78-90, doi:10.1109/MRA.2012.2182810
11. I.D. García-Santillán, M. Montalvo, J.M. Guerrero, G. Pajares. Automatic detection of curved and straight crop rows from images in maize fields. Biosyst Eng, 156 (2017), pp. 61-79, doi:10.1016/j.biosystemseng.2017.01.013
12. Geiger, P. Lenz, R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), (2012)
13. F.A. Ghaleb, A. Zainala, M.A. Rassam, A. Abraham. Improved vehicle positioning algorithm using enhanced innovation-based adaptive Kalman filter. Pervasive and Mobile Computing, 40 (2017), pp. 139-155, doi:10.1016/j.pmcj.2017.06.008
14. R. Gonzalez, F. Rodriguez, J.L. Guzman, C. Pradalier, R. Siegwart. Combined visual odometry and visual compass for off-road mobile robots localization. Robotica, 30 (2012), pp. 865-878, doi:10.1017/S026357471100110X
15. Goshtasby, S.H. Gage, J.F. Bartholic. A two-stage cross correlation approach to template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6 (1984), pp. 374-378, doi:10.1109/TPAMI.1984.4767532
16. M. Grella, E. Gil, P. Balsari, P. Marucco, M. Gallart. Advances in developing a new test method to assess spray drift potential from air blast sprayers. Span J of Agric Res 15 (2017), doi:10.5424/sjar/2017153-10580
17. L. Grimstad, P.J. From. Thorvald II - a Modular and Re-configurable Agricultural Robot. IFAC-PapersOnLine, 50 (2017), pp. 4588-4593, doi:10.1016/j.ifacol.2017.08.1005

18. D. Jiang, L. Yang, D. Li, F. Gao, L. Tian, L. Li. Development of a 3D ego-motion estimation system for an autonomous agricultural vehicle. Biosyst Eng, 121 (2014), pp. 150-159, doi:10.1016/j.biosystemseng.2014.02.016
19. M. Kassler. Agricultural Automation in the new Millennium. Comput Electron Agr, 30 (2001), pp. 237-240, doi:10.1016/S0168-1699(00)00167-8
20. J.P. Lewis. Fast Template Matching. Vis. Interface, 95 (1995), pp. 120-123
21. J. Lindblom, C. Lundström, M. Ljung, A. Jonsson. Promoting sustainable intensification in precision agriculture: review of decision support systems development and strategies. Precis Agric, 18 (2017), pp. 309-331, doi:10.1007/s11119-016-9491-4
22. Mahmood, S. Khan. Correlation-coefficient-based fast template matching through partial elimination. IEEE Transactions on Image Processing, 21 (2012), pp. 2099-2108, doi:10.1109/TIP.2011.2171696
23. MathWorks (2018). Computer Vision System Toolbox
24. H. Moravec. Obstacle Avoidance and Navigation in the Real world by a seeing robot rover. PhD thesis, (1980), Stanford University.
25. N. Nourani-Vatani, J. Roberts, M.V. Srinivasan. Practical Visual Odometry for Car-like Vehicles. IEEE International Conference on Robotics and Automation, 1-7 (2009), pp. 3551-3557, doi:10.1109/ROBOT.2009.5152403
26. D. Scaramuzza, F. Fraundorfer. Visual Odometry Part I: The First 30 Years and Fundamentals. IEEE Robotics & Automation Magazine, 18 (2011), pp. 80-92, doi:10.1109/MRA.2011.943233
27. J. Shi, C. Tomasi. Good features to track. EEE Conference on Computer Vision and Pattern Recognition, (1994), doi:10.1109/CVPR.1994.323794

28. T. Utstumo, F. Urdal, A. Brevik, J. Dørum, J. Netland, Ø. Overskeid et al. Robotic in-row weed control in vegetables. Comput Electron Agr, 154 (2018), pp. 36-45, doi:10.1016/j.compag.2018.08.043
29. E.J. van Henten, C.W. Bac, J. Hemming, Y. Edan. Robotics in protected cultivation. IFAC Proceedings Volumes, 46 (2013), pp. 170-177, doi:10.3182/20130828-2-SF-3019.00070
30. K.A. Vakilian, J. Massah. A farmer-assistant robot for nitrogen fertilizing management of greenhouse crops. Comput Electron Agr, 139 (2017), pp. 153-163, doi:10.1016/j.compag.2017.05.012
31. V. Winkler, B. Bickert. Estimation of the circular error probability for a Doppler-Beam-Sharpening-Radar-Mode. 9th European Conference on Synthetic Aperture Radar, (2012), pp. 368-371
32. J. Yoo, S.S. Hwang, S.D. Kim, M.S. Ki, J. Cha. Scale-invariant template matching using histogram of dominant gradients. Pattern Recognit, 47 (2014), pp. 3006-3018, doi:0.1016/j.patcog.2014.02.016.
33. G. Zaidner, A. Shapiro. A novel data fusion algorithm for low-cost localisation and navigation of autonomous vineyard sprayer robots. Biosyst Eng, 146 (2016), pp. 133-148, doi:10.1016/j.biosystemseng.2016.05.002

5. Thesis conclusion

In this Ph.D. thesis, the development and implementation of innovative and enhanced methodologies for precision agriculture have been presented. The proposed methods fully automate some phases of 3D point cloud processing, such as the automatic detection of the crop rows from the whole model of the considered agricultural environment, opening paths for other crucial processes.

The developed unsupervised algorithm explained in the second chapter is able to automatically cluster and localise individual vine rows within the 3D point clouds models of vineyard and provides information about their spatial layout characterised by vine row end points and a curve following the centre of the row. This information provided by the proposed algorithm can also be used for automated 3D path planning, which is a key task for automation and optimisation of autonomous machines operations in the field. The possibility to automatically cluster and localise vine rows within a 3D point cloud map will lead to a new generation of unsupervised point-cloud processing algorithms aimed at evaluating crop status and developing new procedures for precision agriculture applications.

Whereas in the third chapter an innovative modelling framework has been presented here to generate low complexity 3D mesh models of vine rows from raw 3D point clouds of vineyards. The proposed methodology reduces the number of georeferenced instances required to properly describe the spatial layout and shape of vine canopies; this allows the amount of data to be drastically reduced without losing relevant crop shape information. In addition, the developed algorithm semantically interprets the 3D model by automatically classifying the points of the could in two groups: one representing the vine canopy and the other the terrain. The reduction of the amount of data is a crucial factor in facilitating shorter computational times of huge datasets, such as crop raw 3D point clouds, thus enabling the exploitation of point clouds information in real time operations in the field. The robustness of the proposed algorithms was verified on different vineyard parcels characterised by a sloped land formation with varying elevations.

Finally, in the fourth chapter an enhanced image processing algorithm for a cost-effective monocular visual odometry system, aimed at obtaining highly reliable results at low computational costs for a tracked UGV navigation in agricultural applications, has been explained. The algorithm allows the VO system to be applied to real-time applications using cost-effective hardware, by requiring a lower computational load. The robustness of the proposed VO algorithm was evaluated by performing an extensive in-field test campaign on several terrains typical of agricultural scenarios: soil, grass, concrete, asphalt and gravel.

When considering scenarios involving cooperating machines, data reduction is relevant for enabling fast communication and data exchange between in field actors. The information provided by the proposed methodologies are of vital importance for the interpretation of complex 3D point cloud models of agricultural environment, moving from a macro level (field parcel) to a micro level (plants, fruits, branches) and for infield autonomous machines 3D path planning to complete infield tasks with high accuracy.

6. Publication list

During the Ph.D. the following research works were published;

1. L. Comba, S. Zaman, A. Biglia, D. Ricauda Aimonino, P. Gay (2020). 3D point clouds density-based segmentation method for vine rows localisation / detection. Biosystems Engineering. (**Submitted**)
2. L. Comba, S. Zaman, A. Biglia, D. Ricauda Aimonino, F. Dabbene, P. Gay (2020). Semantic interpretation and complexity reduction of 3D point clouds of vineyards. Biosystems Engineering (doi.org/10.1016/j.biosystemseng.2020.05.013)
3. S. Zaman, L. Comba, A. Biglia, D. Ricauda Aimonino, P. Barge, P. Gay (2019). Cost-effective visual odometry system for vehicle motion control in agricultural environments. Computers and Electronics in Agriculture (doi.org/10.1016/j.compag.2019.03.037)