

3D point cloud density-based segmentation for vine rows detection and localisation



A. Biglia ^a, S. Zaman ^b, P. Gay ^a, D. Ricauda Aimonino ^{a,1}, L. Comba ^{a,c,*}

^a Department of Agricultural, Forest and Food Sciences (DiSAFA) – Università degli Studi di Torino, Largo Paolo Braccini 2, 10095 Grugliasco (TO), Italy

^b Dimensions Agri Technologies AS, Berghagan 3, 1405 Langhus, Norway

^c CNR-IEIIT – Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

Keywords:

Smart agriculture
Precision viticulture
Remote sensing from UAV
Photogrammetry
3D data processing

ABSTRACT

The adoption of new sensors for crop monitoring is leading to the acquisition of large amounts of data, which usually are not directly usable for agricultural applications. The 3D point cloud maps of fields and parcels, generated from remotely sensed data, are examples of such big data, which require the development of specific algorithms for their processing and interpretation, with the final aim to obtain valuable information about crop status.

This manuscript presents an innovative 3D point cloud processing algorithm for vine row detection and localisation within vineyard maps, based on the detection of key points and a density-based clustering approach. Vine row localisation is a crucial phase in the interpretation of the complex and huge 3D point clouds of agricultural environments, which makes it possible to move the focus from a macro level (parcel and plot scale) to a micro level (plants, fruits and branches). The algorithm outputs fully describe the spatial location of each vine row within the whole 3D model of the agricultural environment by a set of key points and an interpolating curve. The algorithm is specifically conceived to be robust and: (i) independent of the adopted airborne sensor used to acquire the in-field data (not requiring a model with colour or spectral information); (ii) able to manage vineyards with any vine row layout or orientation (such as curvilinear) and (iii) not hindered by the occurrence of missing plants. The experimental results, obtained by processing the models of seven case study parcels, proved the algorithm's reliability and accuracy: the automatic vine row detection was found to be 100% in accordance with the manual one; and the obtained localisation indices showed an average error of 12 cm and standard deviation of 10 cm, which is fully compatible with the considered agricultural applications. In addition, the algorithm outputs can be profitably exploited for enhanced path planning of autonomous agricultural machines adopted for in-field operations.

1. Introduction

Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs), also named drones, are assuming a key role in the modern farming approach known as Agriculture 4.0 (Michels et al., 2020; Mammarella et al., 2022a). Indeed, the drones capability to autonomously perform in-field operations is being profitably exploited in many agricultural scenarios (Peng and Vougioukas, 2020; Thompson and Puntel, 2020). The agricultural tasks that currently benefit, or might benefit in the near future, from the adoption of autonomous ground and aerial drones can be grouped into two main categories: crop monitoring

tasks 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; Comba et al., 2021a; Feng et al., 2020), 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), and disease or stresses detection (Kerkech et al., 2020; Guidoni et al., 2021; Comba et al., 2021b). Concerning 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 and Vigneault, 2017).

* Corresponding author at: Department of Agricultural, Forest and Food Sciences (DiSAFA) – Università degli Studi di Torino, Largo Paolo Braccini 2, 10095 Grugliasco (TO), Italy.

E-mail address: lorenzo.comba@unito.it (L. Comba).

¹ Co-last author.

Notation table	
$\mathcal{B}_{x,y}$	cylindrical regions of point cloud $S_1^{\{\text{Main}\}}$, centred in $[xy]^\top$ and with vertical axis
$\mathcal{B}'_{x,y}(\vartheta)$	section of cylindrical region $\mathcal{B}_{x,y}$ at angle ϑ
$c_j^{(\text{RF})}$	j-th canopy central point, with $c_j \in \mathcal{V}_m$ represented in the {RF} Reference Frame
\mathcal{C}	comprehensive set of canopy centre points c
$d_v(x,y)$	local inter row width in $[xy]^\top$
$D_y(\mathcal{B}'_{x,y}, s)$	normalised frequencies distribution histogram of points $p_i \in \mathcal{B}'_{x,y}$ along the $y^{\{\text{Loc}\}}$ axis
$\mathcal{E}_j^{[h]}$	subset of points c_j within an elliptic region of interest at algorithm iteration [h]
E_r	ordered set of key points e representing vine row r
$\{e_0, e_n\}$	key-points couple representing vine row r ends
$e^{[h]}$	enhanced key point e defined at algorithm iteration [h]
g	generic point of the 3D point cloud S_1
$[h]$	Clustering algorithm iteration
$[k]$	Key point detection algorithm iteration
l_r	number of centre points representing r -th vine row
$n_{\mathcal{C}}$	overall number of detected central points of set \mathcal{C}
$n_{\mathcal{V}}$	overall number of detected vine rows
n_r	number of key-points representing the r -th vine row, provided by the algorithm
m_r	number of reference points manually defined for the r -th vine row
$O_{\text{Loc}}^{\{\text{Main}\}}$	origin of local reference frame {Loc} in reference frame {Main}
p_i	projection of key-point e_i on the
$r_{\mathcal{B}}$	radius of the cylindrical subset $\mathcal{B}_{x,y}$
$R_{\text{Loc}}^{\{\text{Main}\}}$	rotation matrix from {Loc} to {Main} reference frame
$S_1^{\{\text{Main}\}}$	3D point cloud model represented in the reference frame {Main}
s	bin of D_y histogram
S_y	set of all the histogram bins
t	distance threshold between key points
\hat{v}_j	j-th local maximum of the density histogram D_y
\mathcal{V}_r	r-th cluster of central points representing vine rows r-th
$\mathcal{V}^{[h]}$	cluster of central points \mathcal{V} at clustering algorithm iteration [h]
$x^{\{\text{RF}\}}$	x axis of the {RF} Reference Frame
$y^{\{\text{RF}\}}$	y axis of the {RF} Reference Frame
$z^{\{\text{RF}\}}$	z axis of the {RF} Reference Frame
<i>Greek letters</i>	
$\vartheta_{v(x,y)}$	angle defining the local vine row direction in $[xy]^\top$
$\vartheta_{\perp v}$	angle defining the local vine row perpendicular direction
δ_s	bin width of histogram D_y

The effectiveness of the adoption of drones for precision agriculture applications is strictly related to the proper knowledge of the working scenarios, in terms of both spatial layout (Chen et al., 2020; Gao et al., 2020) and crops status and needs. Indeed, the joint contribution of such information allows to reach the target and/or to properly modulate agronomical operations in a timely fashion (Gil et al., 2013), ensuring, at the same time, the required safety (Wang et al., 2020). In this context, to properly achieve the tasks, the accurate and reliable path planning and control strategies of the drones are thus essential (Dusadeerungsikul and Nof, 2019; Khajepour et al., 2020), which involve crucial information such as parcel boundaries and crop location (Graf Plessen and Bemporad, 2017; Li et al., 2020).

Sensors that are able to provide 3D models of the agricultural environment can lead to favourable improvements in the description of the complex scenarios in which the drones operate (Chakraborty et al., 2019; Comba et al., 2019). 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., 2022), branches location for automatic pruning (Cuevas-Velasquez et al., 2020), and crop location for accurate path planning (Jurado et al., 2020; Malavazi et al., 2018). Such 3D models are usually in the form of a point cloud, which is a set of unordered points in the 3D space. A 3D point cloud can be derived by using Structure from Motion (SfM) algorithms (Gené-Mola et al., 2020), Light Detection and Ranging systems (LiDAR) (Blanquart et al., 2020), or depth cameras (Condotta et al., 2020). However, specific algorithms have to be developed to properly extract valuable crop information from raw 3D models (Escolà et al., 2017; Sanz et al., 2018), also exploiting artificial intelligence tools (Zhang et al., 2021). In this process, an essential phase of the processing algorithms is usually the semantic interpretation and segmentation of the 3D point clouds, which assign each point to different portions of the whole model. Indeed, many applicative examples of semantic interpretation of 3D point cloud for agricultural purposes can be found in literature, aimed at detecting portion of the model representing leaves, branches, fruits, buds, and other elements (Díaz et al., 2018; Mortensen et al., 2018; Zhou

et al., 2019; Zeng et al., 2020; Comba et al., 2020a). These 3D point cloud algorithms have been usually developed to process a single tree, or portion of a crop, at a time (Comba et al., 2020b; Gené-Mola et al., 2020; Zhang et al., 2021). 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 whole considered agricultural environment is thus required. This is a crucial phase in the interpretation of the complex and huge 3D point clouds of the agricultural environments, which focuses from a macro level (parcel and plot scale) to a micro level (plants, fruits, branches).

This manuscript presents a 3D point cloud processing algorithm for vine row detection and localisation within vineyard maps, based on the detection of key points and a density-based clustering approach. The proposed method is completely new, and no methodologies for this specific objective have been previously proposed in literature. The algorithm provides as an output an ordered set of key points in a 3D coordinate system representing the canopy's central points (which includes the two vine row end points) and a curve representing the spatial layout of each vine row in the map. The peculiarities of the 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).

The unsupervised detection of each single vine row within a whole vineyard model is essential not only to enable the automatic application of further specific processing algorithms at plant scale, but also for in-field path planning purposes. Indeed, 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 (Mammarella et al., 2020). By overcoming 2D path planning algorithms, 3D path planning fully exploits the terrain's and environmental characteristics (Jin and Tang, 2011; Hameed et al., 2016; Mammarella et al., 2022a, 2022b).

This manuscript is structured as follows: Section 2 describes the experimental field and the acquisition campaigns; Section 3 presents the

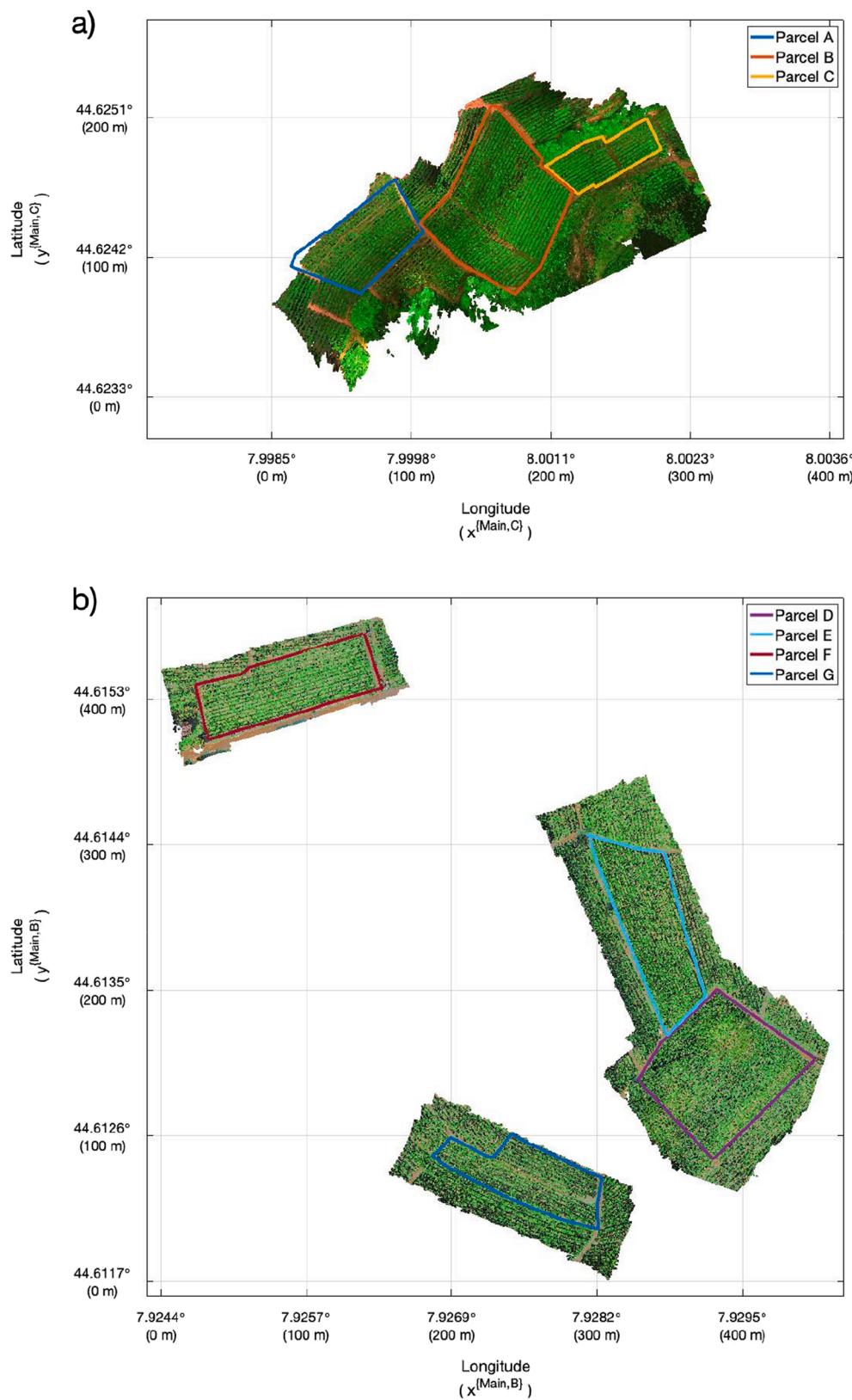


Fig. 1. 3D point clouds of the seven parcels considered as a case study, located in Serralunga d'Alba - Italy (a) and in Barolo - Italy (b).

innovative algorithm for the vine rows localisation and clustering; the results are presented and discussed in [Section 4](#), while the conclusions are reported in [Section 5](#).

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, Northwestern Italy), was considered as a case study and was modelled by dense 3D point clouds. The considered parcels have extensions ranging from 0.17 to 0.71 ha, covering an overall surface of about 3.4 ha, and they are characterised by a sloped land conformation. Six parcels (named A, B, D, E, F and G) are cultivated with the Nebbiolo vine variety and one parcel (named C) is cultivated with the Moscato vine variety ([Fig. 1](#)), using a vertical shoot position trellis system. The parcels consist of 21, 33, 13, 35, 21, 16 and 16 vine rows, respectively. The space between the vine plants and the inter-row space was about 0.9 m and 2.5 m in all the considered parcels.

The 3D point clouds were obtained by using the Agisoft Photoscan® software (2020, St. Petersburg, Russia), which is based on the SfM algorithms, and by processing the UAV-based aerial images. The seven point clouds are made by 2.4 up to 10.2 million points and were saved in individual.las files with sizes ranging from about 44 Mb to 185 Mb, respectively. The mean density of the generated 3D point clouds was about 1450 points per m^2 of the map surface. 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 occurred 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 planning a set of waypoints, which were defined on the basis of the vineyards geographic information system map with the drone mission planning and flight control software (UgCS, 2021). A forward and side overlap greater than 80% was guaranteed between adjacent images to help the images alignment process. Prior to the alignment of the images, 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 the vineyards used in this study had neither colour nor spectral information, so that only the spatial information provided by each point was exploited by the proposed processing method. This approach was introduced to make the proposed processing algorithm unaffected by the type of airborne sensor or by the spectral difference that could characterise different vineyard environments during the growing season.

3. 3D point cloud processing method

The developed algorithm, which automatically detects and localises each single vine row within the 3D point clouds of the vineyards, has two main outputs: (i) an ordered set of key points representing the canopy's central points in a 3D coordinates system (which includes the two vine row end points) and (ii) a curve representing the spatial layout of each vine row in the map. The vine rows detected within the 3D point cloud are automatically sorted and numbered (from southwest to northeast), and the algorithm also provides their length, altitude difference and average orientation with respect to the west to east direction.

The algorithm, which is based on a sequence of operations sketched in the flowchart in [Fig. 2](#), can be divided into three main phases: (1) the detection of a set of points representing the centre of the vine canopies, (2) the clustering of the central points in groups, each one representing a single vine row, and, finally, (3) the sorting and refinement procedure to define a set of ordered key points, essential to determine the curves describing the location of each vine row. The specific peculiarities that characterise the 3D point clouds of the vineyards, which prevent the adoption of already available processing algorithms, are discussed in detail in each processing phase, together with the innovative solutions that have been envisioned. Please note that, in the here presented new point cloud processing method, some specific steps of the algorithm

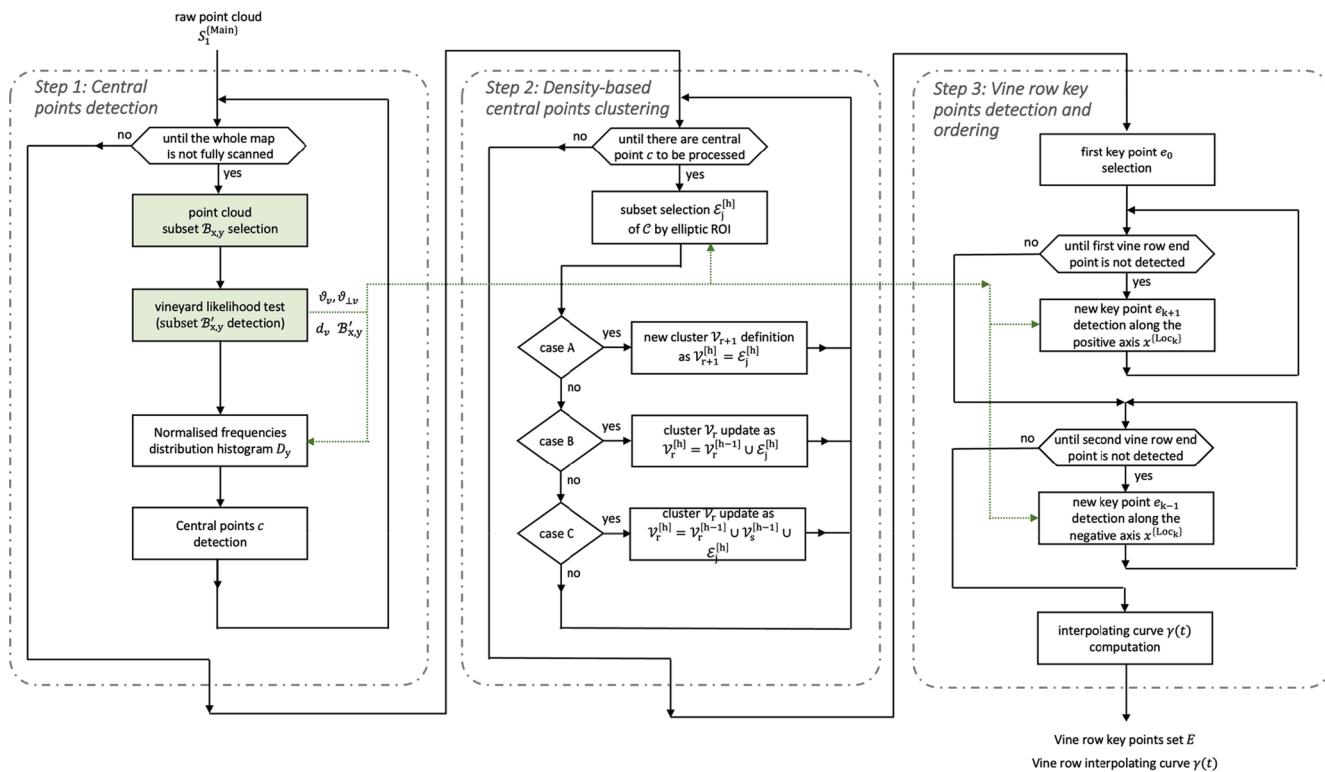


Fig. 2. Flow chart of the developed point cloud processing settlement. Operations and outputs defined in [Comba et al. \(2018\)](#), exploited in the proposed new method, has been highlighted in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

presented in Comba et al. (2018) (the aim of which is the detection of parcel boundaries) have been exploited. These processing steps are represented by the two green highlighted blocks of the flowchart in Fig. 2.

3.1. Central points detection

The first phase of the algorithm is the detection of a set of points c_j , representing the canopy central points of the vine rows, which is obtained by processing the raw dense 3D point cloud of the vineyard. This

first step allows the main information of the canopy, required in the second and third steps, to be extracted from the raw 3D point cloud. In addition, this processing phase reduces the amount of model instances to be processed in the subsequent steps and, thus, speeds up the second and third computing phases. This processing phase, named *Step 1*, is represented in the left part of the flowchart in Fig. 2. The central points were detected by processing several subsets of the whole vineyard 3D model, which were defined by a scanning approach (also called *mobile window* approach). This approach, firstly introduced in Comba et al. (2018), in this work has been significantly updated as described below.

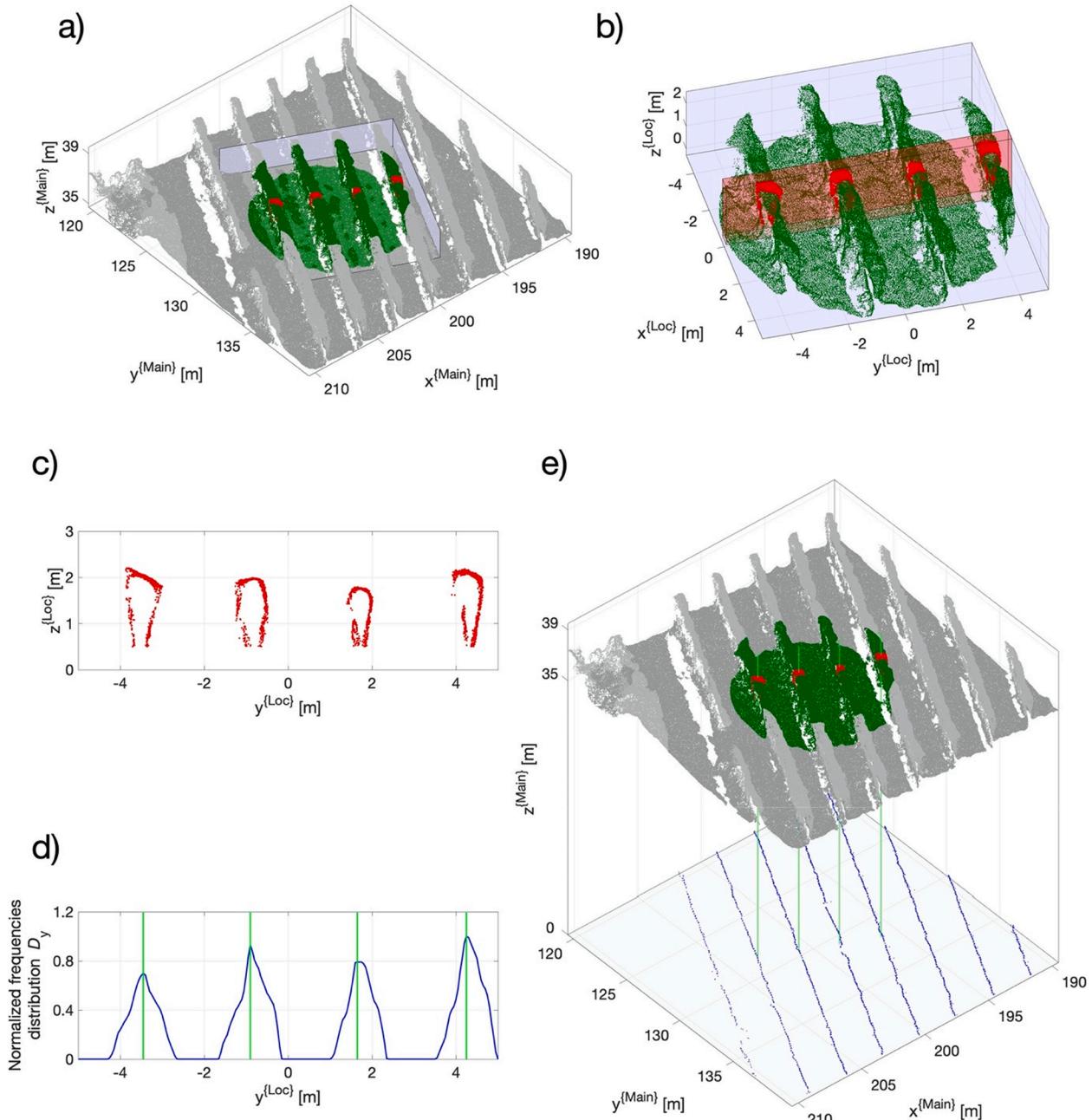


Fig. 3. (a) Cylindrical subset $\mathcal{B}_{x_b, y_b}^{\{Main\}}$ (green dots) of the 3D point cloud $S_1^{\{Main\}}$ (grey dots) represented in the Main Reference Frame $\{Main\}$; (b) enlargement of the subset $\mathcal{B}_{x_b, y_b}^{\{Loc\}}$ (green dots) represented in the Local Reference Frame $\{Loc\}$ (with origin in $O_{Loc}^{\{Main\}} = [x_b, y_b, z_b]^T = [200, 130, 35]^T$ m) and subset $\mathcal{B}'_{x_b, y_b}(\vartheta_{\perp y})$ (red dots), determined with $\vartheta_{\perp y} = 1.1\pi$; (c) projection of subset $\mathcal{B}'_{x_b, y_b}(\vartheta_{\perp y})$ on the $yz^{\{Loc\}}$ plane (as in Comba et al., 2018); (d) normalised frequencies distribution histogram $D_y(\mathcal{B}'_{x_b, y_b}, s)$ (blue line) with detected local maxima \hat{v}_j (green line); (e) detected central points $c_j^{\{Main\}}$ (green line) of the canopy in the original 3D point cloud $S_1^{\{Main\}}$ (grey dots) and the entire set of central points of canopy C (blue dots). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In Comba et al. (2018), with the aim to identify the vineyard parcels boundaries, the raw 3D point cloud ($S_1^{\{\text{Main}\}}$) was scanned with a *mobile window* by selecting different cylindrical regions (named $\mathcal{B}_{x,y}$), centred in $[xy]^\top$ and with radius $r_{\mathcal{B}}$. The scanning procedure proceeds with a vineyard likelihood test, which is performed on each cylindrical region $\mathcal{B}_{x,y}$ and provides the following outputs: (1) the angles θ_v and $\theta_{\perp v}$, which identify the parallel and perpendicular direction with respect to the *local* vine row orientation, (2) the *local* inter row distance d and (3) the subset section $\mathcal{B}'_{x_b,y_b}(\theta_{\perp v})$. With the term *local*, the centre of the cylindrical region \mathcal{B} was here considered. For a complete discussion about this procedure, please refer to Comba et al. (2018). In Fig. 2, this processing phase is represented by the two green highlighted blocks of the flowchart, and its outputs are exploited in some specific subsequent phases, as represented by the green arrows. An example of the outputs of this procedure is reported in Fig. 3a-3c. In particular, the cylindrical region \mathcal{B}_{x_b,y_b} (centred at $[x_b, y_b]^\top = [200, 130]^\top$ m with a radius $r_{\mathcal{B}} = 5$ m), is highlighted in Fig. 3a (green dots) within the surrounding 3D model (grey dots) and enlarged in Fig. 3b. Also, the subset $\mathcal{B}'_{x_b,y_b}(\theta_{\perp v})$ was reported in Fig. 3b (red dots), which was selected by using the angle value $\theta_{\perp v} = 1.1\pi$. The angle θ is measured anticlockwise from the $x^{\{\text{Main}\}}$ axis on the horizontal $xy^{\{\text{Main}\}}$ plane.

In the processing algorithm presented in this work, subset sections $\mathcal{B}'_{x,y}(\theta_{\perp v})$ are here exploited in a new way by detecting the canopy central points c . In particular, this was performed computing and analysing the density distribution of the points p within section $\mathcal{B}'_{x,y}(\theta_{\perp v})$ which, for definition, intersects several vine rows. Indeed, the angle $\theta_{\perp v}$ used for section $\mathcal{B}'_{x,y}(\theta_{\perp v})$ detection, represents the perpendicular direction with respect to the local vine row orientation. To detect the central points c , a Local Reference Frame {Loc} was introduced, whose origin is located in $O_{\text{Loc}}^{\{\text{Main}\}} = [x_b, y_b, z_b]^\top$ (centre of the cylindrical region) and with the $x^{\{\text{Loc}\}}$ and $y^{\{\text{Loc}\}}$ axes parallel and perpendicular to the vine row respectively. The coordinate z_b was chosen to be equal to the local elevation of the digital terrain model. The density of the 3D point subset was thus assessed by computing the normalised frequencies distribution histogram of the points g along the $y^{\{\text{Loc}\}}$ axis.

$$D_y(\mathcal{B}'_{x,y}, s) = \text{card}\{g = [x, y, z]^\top \in \mathcal{B}'_{x,y}(\theta_{\perp v}) : |y - s| < \frac{\delta_s}{2}\} \cdot \text{card}(\mathcal{B}'_{x,y}(\theta_{\perp v}))^{-1} \quad (1)$$

with $s \in S_y = \{-r_{\mathcal{B}}, -r_{\mathcal{B}} + \delta_s, -r_{\mathcal{B}} + 2\delta_s, \dots, 0, \dots, r_{\mathcal{B}}\}$, where S_y is the set of all the histogram bins and δ_s is the bins width. In Fig. 3c and 3d, the sample subset of points $\mathcal{B}'_{x_b,y_b}(\theta_{\perp v})$ and its normalised frequencies distribution histogram are represented. More in detail, the position of $c_j^{\{\text{Loc}\}}$ was determined by detecting the local maxima \hat{v}_j in the density histogram $D_y(\mathcal{B}'_{x,y}, s)$, as:

$$c_j^{\{\text{Loc}\}} = [0, \hat{v}_j, 0]^\top \quad (2)$$

The central points $c_j^{\{\text{Loc}\}}$ of the canopy are here represented in the Local Cartesian Reference Frame {Loc}, and their absolute position in the Main Reference Frame {Main} was simply retrieved as follows.

$$\begin{aligned} c_j^{\{\text{Main}\}} &= \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = R_{\text{Main}}^{\text{Loc}} \cdot c_j^{\{\text{Loc}\}} + O_{\text{Loc}}^{\{\text{Main}\}} \\ &= \begin{bmatrix} \cos\theta_{\perp v} & -\sin\theta_{\perp v} & 0 \\ \sin\theta_{\perp v} & \cos\theta_{\perp v} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \hat{v}_j \\ 0 \end{bmatrix} + \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \end{aligned} \quad (3)$$

where $R_{\text{Main}}^{\text{Loc}}$ is the rotation matrix from {Loc} to {Main} and $O_{\text{Loc}}^{\{\text{Main}\}}$ is the {Loc} origin, expressed in the {Main} coordinates. For example, the detected local maxima \hat{v}_j in the density histogram $D_y(\mathcal{B}'_{x,y}, s)$ are reported in green in Fig. 3d, while the detected vine row centres $c_j^{\{\text{Main}\}}$ are highlighted in green in Fig. 3e. In the example of Fig. 3, 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, which led to the four canopy central points $c_1^{\{\text{Main}\}} = [203.13, 128.55, 35.03]^\top$, $c_2^{\{\text{Main}\}} = [200.82, 129.62, 35.08]^\top$, $c_3^{\{\text{Main}\}} = [198.50, 130.69, 35.09]^\top$ and $c_4^{\{\text{Main}\}} = [196.14, 131.78, 35.07]^\top$ metres.

By applying the procedure described above to an entire 3D point cloud, a set of central points $\mathcal{C} = \{c_j, j = 1, \dots, n_{\mathcal{C}}\}$ of all the vineyard canopies can be obtained, where $n_{\mathcal{C}}$ is the overall number of computed central points. To simplify the discussion of the next steps of the algorithm, the projection of the central points \mathcal{C} on the 2D plane $z^{\{\text{Main}\}} = 0$ will be considered, as shown in Fig. 3e with blue dots. All the results can be reported in the original 3D system by restoring the DTM, considering the local terrain elevation as the z coordinate of the points. The results obtained by processing the entire 3D point cloud S_1 of Fig. 1a are reported in Fig. 4.

The set of canopy central points \mathcal{C} obtained by the proposed new method, together with the additional information regarding the vineyards local features provided by Comba et al. (2018) (the local vine row direction $\theta_v(x, y)$ and of the local inter row spacing $d_v(x, y)$, which can be properly organised in maps) will be used in the following algorithm phases, called *Step 2* and *Step 3* (Fig. 2).

3.2. Density-based central points clustering

The obtained set \mathcal{C} of the canopy central points is, in this phase, still unordered and unclustered. Indeed, the central points provided by the previous processing phase are not associated to any specific vine row and their position within the vine row is not related to their generation order. To detect the location of each vine row, a clustering and sorting procedure of the central points is thus required. The output of this second phase of the algorithm is indeed a set of point clusters \mathcal{V}_r , with $r = 1, \dots, n_{\mathcal{V}}$, and where $n_{\mathcal{V}}$ is the number of the detected vine rows. This processing phase, named *Step 2*, is represented in the central part of the flowchart in Fig. 2. It should be noted that the points within 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). However, the occurrence of missing and/or diseased plants leads to vine row interruptions that, in some cases, can be more extended than the inter-row path dimension. Indeed, considering an inter-row path ranging between 2 and 3 m and with a vine plants distance ranging from 0.9 to 1.2 m, as shown in Fig. 5a, the occurrence of two consecutive missing and/or diseased plants can lead to the absence of key points in a portion of the vine row longer than the inter-row path; this causes all the available 3D point cloud clustering methods (such as fuzzy clustering, partitioning methods, hierarchical clustering or standard density-based algorithms), to fail (Biosca and Lerma, 2008; Chen et al., 2022; Li et al., 2022).

The proposed enhanced density-based clustering approach performs an iterative clustering procedure which defines small subsets of points representing portions of vine rows, and merging them sequentially when specific criteria are fulfilled. Considering the algorithm iteration [h], a subset $\mathcal{E}_j^{[h]}$ of points centred in c_j and within an elliptic Region of Interest (ROI) was defined as.

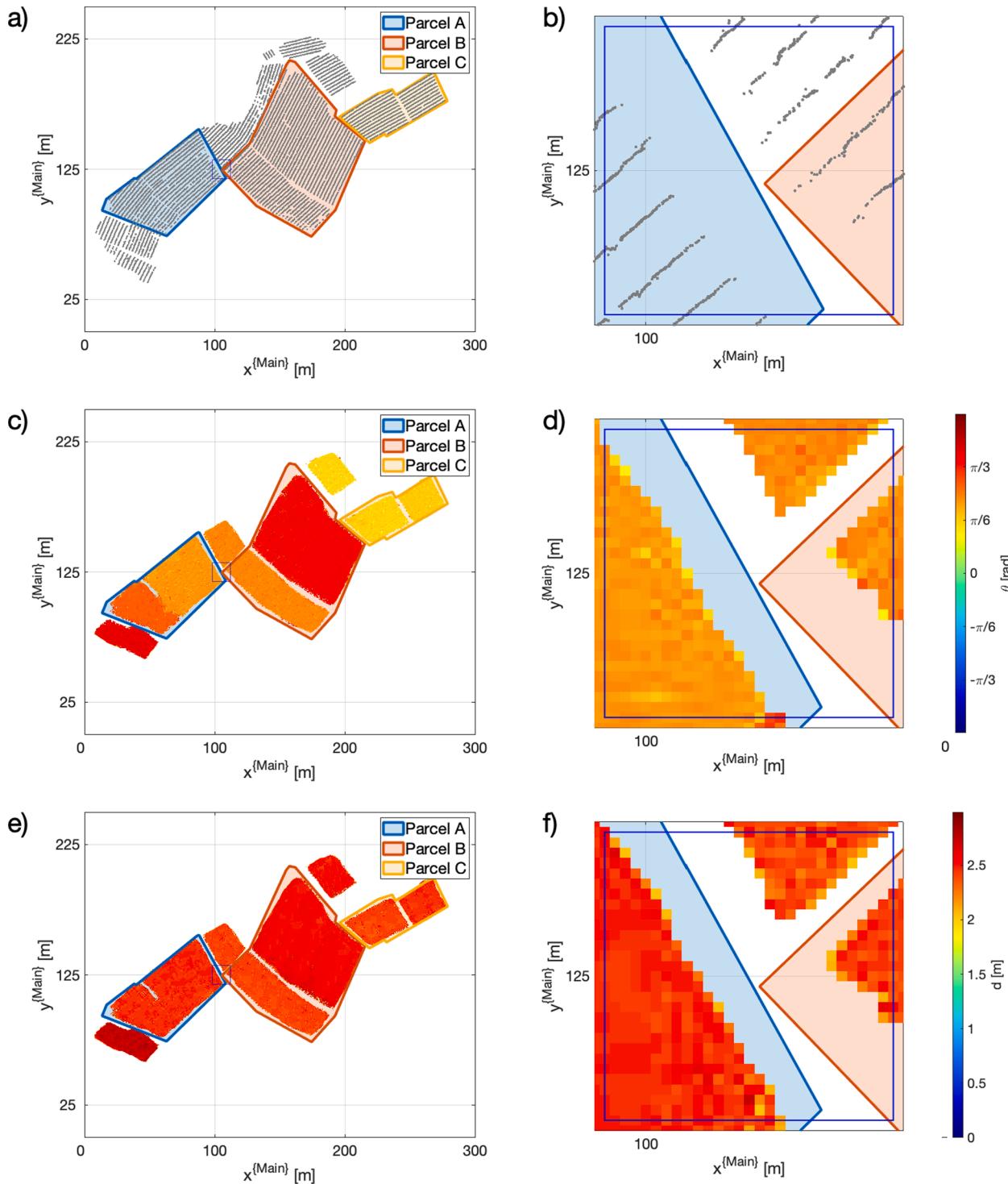


Fig. 4. (a) the set of canopy centre points \mathcal{C} , (c) the map of the local vine row direction $\theta_v(x,y)$ and (e) the map of the local inter row spacing $d_v(x,y)$, obtained by processing the entire point cloud S_1 of Fig. 1a, and their enlargements (b), (d) and (f), respectively. Variables θ_v and d_v are outputs defined in Comba et al. (2018).

$$\mathcal{C}_j^{[h]} = \left\{ [x, y, z]^T \in \mathcal{C}^{\{Main\}} \mid \frac{(x \cdot \cos \theta_v + y \cdot \sin \theta_v - x_j)^2}{3 \cdot d_v} + \frac{(-x \cdot \sin \theta_v + y \cdot \cos \theta_v - y_j)^2}{0.5 \cdot d_v} \leq 1 \right\} \quad (4)$$

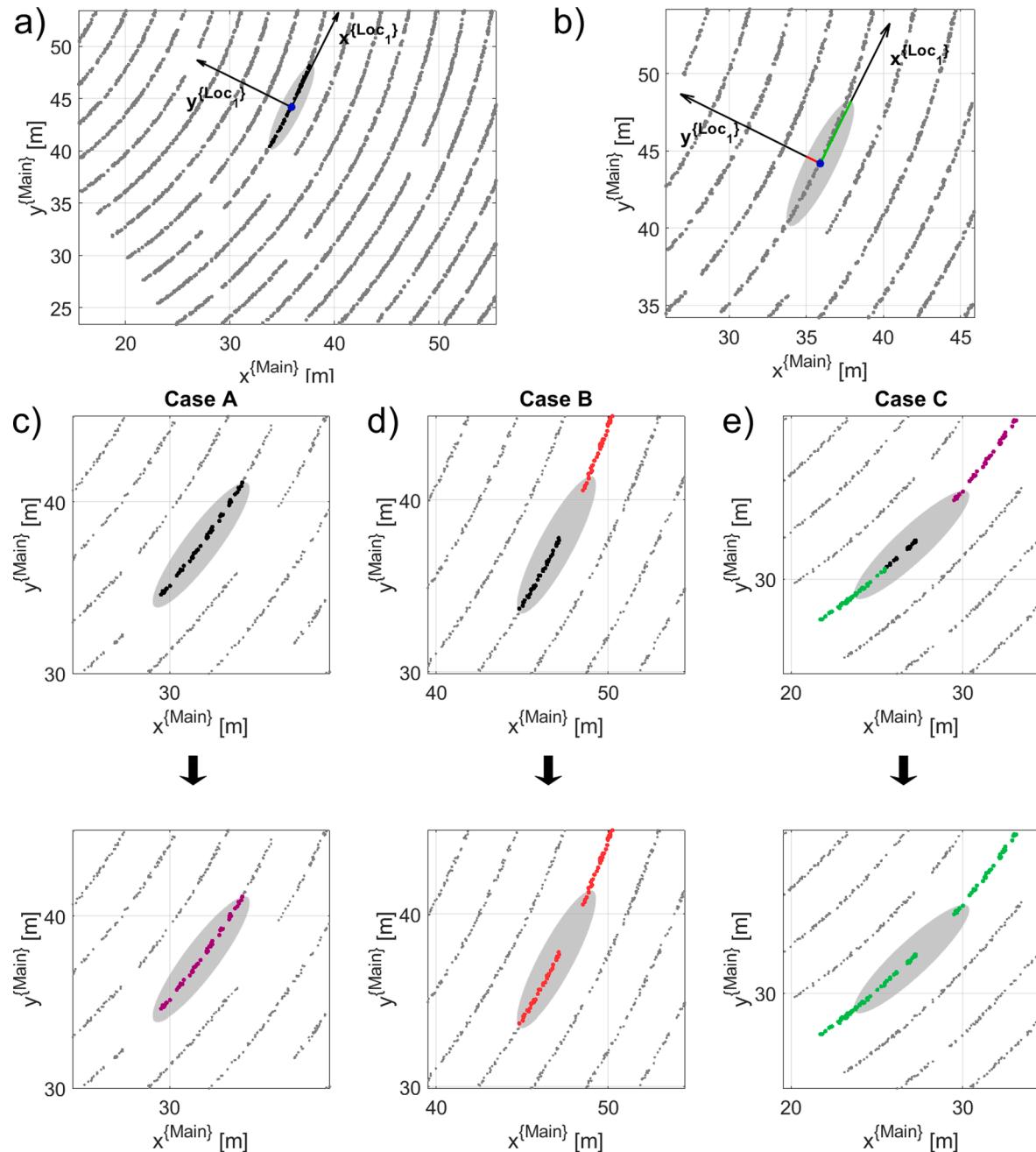


Fig. 5. (a) the entire set of the canopy central points \mathcal{C} (grey dots) in the Local Reference Frame $\{\text{Loc}_j\}$, and an example of the elliptic ROI (grey area) that is used to select the neighbouring points \mathcal{E}_j (black dots) of c_j (blue dot); (b) enlargement of subplot (a), with the highlighted major (green line) and minor (red line) semi axes of the ROI; Case A (c), Case B (d) and Case C (e) scenarios of the clustering process. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where (x_j, y_j) are the coordinates of the central point c_j , while d_v and θ_v are the local inter-row width and vine row orientation in (x_j, y_j) . The values of d_v and θ_v parameters, which are outputs of the method presented in Comba et al. (2018), have been derived from specific maps. In Fig. 4b and 4c, the sample maps of the local vine row direction $\theta_v(x, y)$ and the local inter row spacing $d_v(x, y)$, obtained by processing the entire point cloud S_1 , are reported. Alternatively, the neighbourhood $\mathcal{E}_j^{[h]}$ can also be defined as.

$$\mathcal{E}_j^{[h]} = \left\{ [x, y, z]^\top \in \mathcal{C}^{\{\text{Loc}_j\}} \mid \frac{x^2}{3 \cdot d_v} + \frac{y^2}{0.5 \cdot d_v} \leq 1 \right\} \quad (5)$$

if the Local Reference Frame $\{\text{Loc}_j\}$ (defined with its origin in c_j and with axes $x^{\{\text{Loc}_j\}}$ and $y^{\{\text{Loc}_j\}}$ tangent and perpendicular to the local vine row orientation θ_v , respectively) is considered. An example of the points $\mathcal{E}_j^{[h]}$ selected with the elliptic ROI is represented in Fig. 5a, while its enlargement is reported in Fig. 5b.

Then, depending on the status of points belonging to $\mathcal{E}_j^{[h]}$, four scenarios of cluster assignment can occur. In the first scenario (case A), when the points within $\mathcal{E}_j^{[h]}$ do not belong to any previously defined vine

row cluster $\mathcal{V}_r^{[h-1]}$, a new cluster $\mathcal{V}_{r+1}^{[h]} = \mathcal{E}_j^{[h]}$ is defined. In the second scenario (case B), at least one point within the subset $\mathcal{E}_j^{[h]}$ belongs to a single, already defined cluster $\mathcal{V}_r^{[h-1]}$ and, thus, $\mathcal{E}_j^{[h]}$ and $\mathcal{V}_r^{[h-1]}$ are merged forming the (updated) cluster $\mathcal{V}_r^{[h]} = \mathcal{V}_r^{[h-1]} \cup \mathcal{E}_j^{[h]}$. The third scenario (case C) is an extension of case B and occurs when some (or even all) the points within $\mathcal{E}_j^{[h]}$ belong to more than one cluster, such as $\mathcal{V}_r^{[h-1]}$ and $\mathcal{V}_s^{[h-1]}$. In this case, all the clusters are merged with the oldest one, together with $\mathcal{E}_j^{[h]}$, updating $\mathcal{V}_r^{[h-1]}$ as $\mathcal{V}_r^{[h]} = \mathcal{V}_r^{[h-1]} \cup \mathcal{V}_s^{[h-1]} \cup \mathcal{E}_j^{[h]}$ and discharging the cluster $\mathcal{V}_s^{[h-1]}$. Please note that the merging procedure performed in clustering cases B and C leads to growing clusters, which are no more bounded by an elliptic ROI. The fourth scenario (case D) concerns a subset $\mathcal{E}_j^{[h]}$ with all the points belonging to a single, already defined cluster $\mathcal{V}_r^{[h-1]}$: in this case, no operations are performed. The iterative procedure is repeated until at each central point c a subset \mathcal{E} has been assigned for the proper clustering task. A graphical representation of three sample clustering iterations (belonging to cases A, B and C, respectively), is reported in Fig. 5.

The elliptic ROI was introduced to improve standard density-based clustering approaches, which exploit circular/spherical ROIs. Indeed,

in order to avoid obtaining clusters with multiple vine rows, the maximum value of the circular ROI radius is limited to a fraction of the inter-row path dimensions. In this way, however, a single vine row could be wrongly represented by many clusters, if missing and/or diseased plants occur. The length of the major semi axis of the ROI allows thus to overcome this limit.

Please note that, during the initial phase of the clustering process (when cases A are more frequent than cases C), the number of overall clusters \mathcal{V} can exceed the number of vine rows $n_{\mathcal{V}}$ to be detected. This trend is reversed during the second part of the clustering process, when the number of clusters \mathcal{V} decreases (as cases C is more frequent than cases A and B), finally settling to the detected vine rows number $n_{\mathcal{V}}$.

3.3. Vine rows key points detection and ordering

With the final aim of defining a continuous curve representing the spatial location of a single vine row, the central points in the cluster \mathcal{V}_r have to be refined. Indeed, no methods exist that can interpolate a set of unordered points in the space, which can be generated by a not injective function (de Boor, 1978; Kong et al., 2021). This is the case of central points representing vine rows with any layout within a map (e.g. curvilinear). A viable solution to this problem can be the adoption of

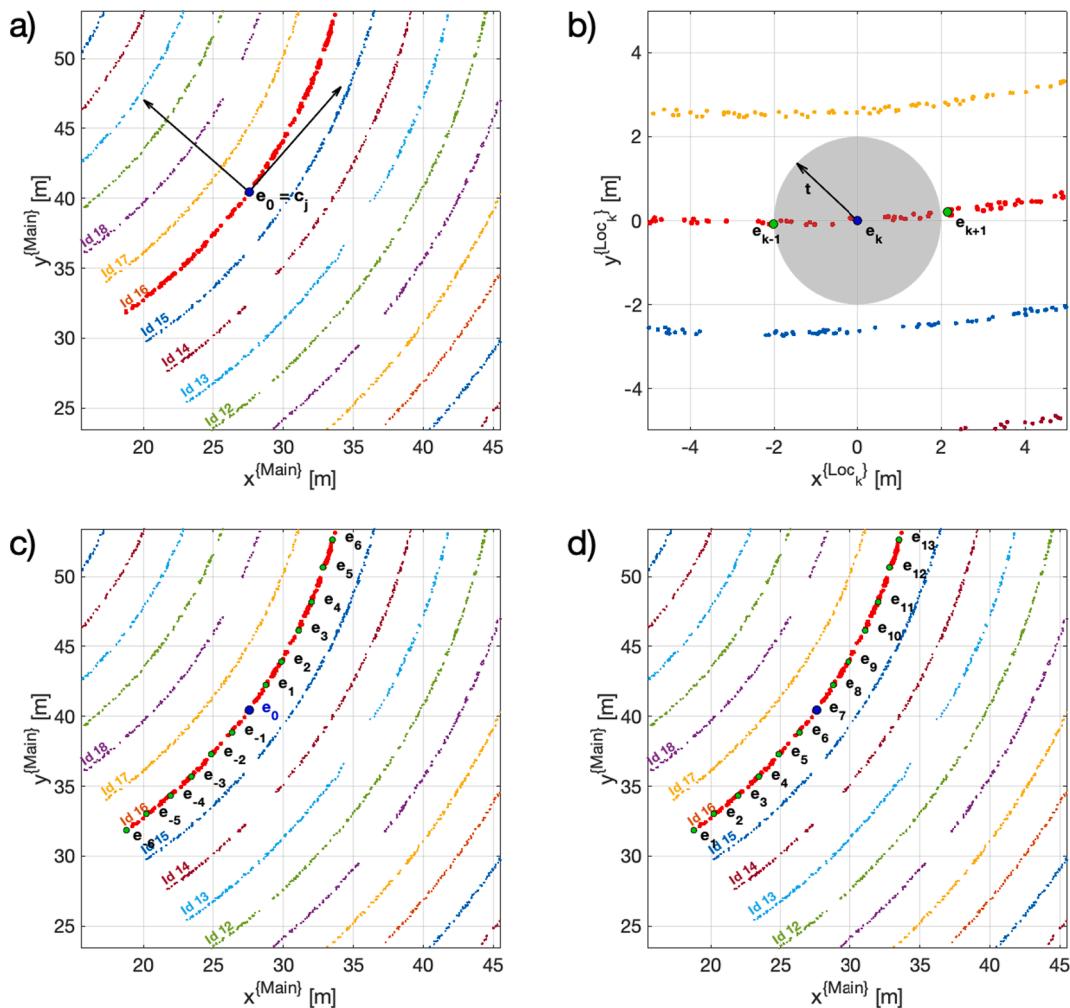


Fig. 6. (a) the set \mathcal{V}_r of the canopy central points representing the vine row $r = 16$ (red dots), the randomly selected first key point $c_j = e_0$ (blue dot), and the Local Reference Frame $\{\text{Loc}_k\}$; (b) sample of new key points e_{k+1} (green dots) detection along the positive direction of axis $x^{[\text{Loc}_k]}$ by using the distance threshold $t = 2$ m (grey area); (c) results of the completed refining and sorting procedure with the detection of vine row end point e_{-6} and (d) final key-points numbering $E_{16} = \{e_1, e_2, \dots, e_{47}\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

recent approaches for interpolating an ordered set of points within the space, which requires the central points to be ordered (D'Errico, 2021). In addition, the general vine row layout prevents the application of a simple ordering procedure of the central points based on a single direction (such as along the x or y axes of the main reference frame).

A refinement of the central points $c \in \mathcal{V}_r$ is introduced in this last phase of the algorithm in order to obtain a set of key points $e \in E_r$ that are ordered, less dense and more equally spaced along the vine row. This processing phase, named *Step 3*, is represented by the right part of the flowchart in Fig. 2. The proposed procedure is based on the following idea: one central point c_j is randomly selected from \mathcal{V}_r and then, starting from it, the vine row is *scanned* towards the two end points. More in detail, the developed algorithm is as follows:

```

k = 0
Randomly selection of the first key point  $e_k = c_j$ 
REPEAT:
Detection of new key point  $e_{k+1}$  as.

$$e_{k+1}^{\{\text{Loc}_k\}} = \arg \min_{c \in \mathcal{V}_m} \|c - e_k\|_2$$


$$\|c - e_k\|_2 > t$$


$$x > 0$$

k = k + 1
UNTIL:  $\{c \in \mathcal{V}_m : \|c - e_k\|_2 > t \vee x > 0\} = \emptyset$ 
Detection of vine row end point  $e_{k+1}$  as.

$$e_{k+1}^{\{\text{Loc}_k\}} = \arg \max_{c \in \mathcal{V}_m} \|c - e_k\|_2$$


$$x > 0$$

k = 0
Repeat:
Detection of new key point  $e_{k-1}$  as.

$$e_{k-1}^{\{\text{Loc}_k\}} = \arg \min_{c \in \mathcal{V}_m} \|c - e_k\|_2$$


$$\|c - e_k\|_2 > t$$


$$x < 0$$

k = k - 1
Until:  $\{c \in \mathcal{V}_m : \|c - e_k\|_2 > t \vee x < 0\} = \emptyset$ 
Detection of vine row end point  $e_{k-1}$  as.

$$e_{k-1}^{\{\text{Loc}_k\}} = \arg \max_{c \in \mathcal{V}_m} \|c - e_k\|_2$$


$$x < 0$$

End

```

An example of this task applied to vine row 16 is reported in Fig. 6. The threshold t was introduced to avoid the selection of a new key point e_{k+1} too close to e_k and its value was set to 2 m to guarantee the generation of key points with a density compatible with the proper description of a vine row with a general layout (also curvilinear). Indeed, this value is usually much lower than the inter-pole distance in trellis systems, which limits the vine row curvature. At each iteration of the algorithm [k], the minimisation problem is solved considering the central point $c \in \mathcal{V}_m$ represented in the Local Reference Frame $\{\text{Loc}_k\}$, introduced and defined in section 3.2, with origin in key point e_k . This allows to simply define the two direction of *scan* of the cluster \mathcal{V}_m towards the two vine row end points by introducing the conditions $x^{\{\text{Loc}_k\}} > 0$ and $x^{\{\text{Loc}_k\}} < 0$, respectively. The algorithm is thus iterated until the two end points of the vine row \mathcal{V}_m are detected, that happens when, at the generic iteration [k], no points c can be found outside the radial threshold distance t . In this case, the threshold check is omitted and the farthest point c from e_k is selected as the end point of the vine row. When both the end points have been detected, the scanning procedure is then considered complete (Fig. 6c). The output of this processing phase for the vine row r is thus an ordered set of n_r key-points $E_r = \{e_1, e_2, \dots, e_{n_r}\}$, with e_1 and e_{n_r} being the two ends of the vine row. An example of the result of this processing phase, applied to sample vine row 16, can be seen in Fig. 6d.

Finally, a unique id number is assigned to each detected vine row within a parcel. Once the ordered set of key points E is defined for each vine row, the curve $\gamma(t)$ interpolating them can be retrieved by the distance-based interpolation function developed by D'Errico (2021) and implemented in Matlab (Fig. 7). Further outputs of the algorithm are the individual vine row length, average orientation angle, and altitude

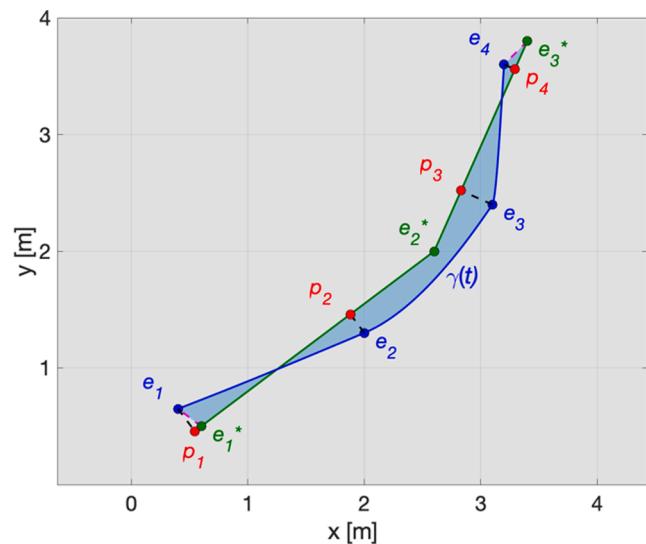


Fig. 7. Exemplificative algorithm output: ordered set of the key-points $E = \{e_1, e_2, \dots, e_4\}$ (blue dots), with the two vine row end-points e_1 and e_4 , and the interpolating curve $\gamma(t)$ (blue line), representing the spatial layout of the vine row. For the algorithm performance assessment: projection e^* of the key-points e on the reference vine row location (green line) defined by the manually generated reference key points p (green dots). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

difference. The points spatial coordinates are represented both in the local and WGS84 reference frames.

3.4. Algorithm performance indices

To assess and quantify the performance of the proposed vineyard 3D point cloud processing method, two families of indices were defined: the first group aimed at properly quantifying the performance of vine rows *detection* (indices 1, 2 and 3) and the second at evaluating the accuracy of the vine rows *localisation* (indices 4, 5 and 6). The indices definitions

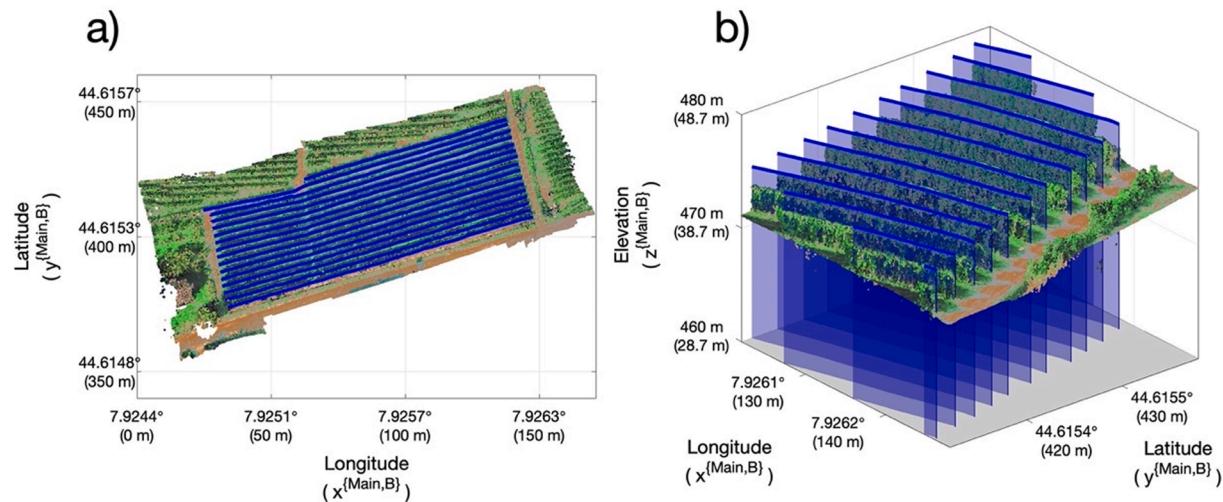
Table 1
Definitions of performance indices used for algorithm accuracy assessment.

	Index name	Definition
Detection indices	1. Good detection	Percentage of properly detected vine rows with respect to the overall number of vine rows in the case study
	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.g. e_1) and manual reference one (e.g. e_1^*): $DEP = \frac{1}{2} \sum_{(i,j) \in \{(1,1), (n_k, n_k)\}} \ e_i - e_j^*\ _2$
	5. Euclidean distances of key points (DEK)	Average distance of all automatically detected key points and the manual reference line: $DEK = \frac{1}{n_k} \sum_{i=1}^{n_k} \ e_i - p_i\ _2$ where n_k is the number of points defining vine row r and p_i is the projection of e_i on the manual reference line (Fig. 7): $COF = \frac{A_r}{\sum_{i=1}^{m_r-1} \ e_{i+1}^* - e_i^*\ _2}$
	6. Curves overlapping factor (COF)	Ratio of the area A_r of the region delimited by algorithms γ_r and manual lines γ_r^* , and the manually detected vine row length (Fig. 7):

Table 2

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 {Main, C} [m, m, m]	End points locations [lat, lon, alt] in {WGS84} [deg, deg, meter]	Total number of enhanced key-points, card (E_r)
1F	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
2F	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
3F	118.55	17.27	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
4F	118.34	17.52	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
5F	118.43	17.65	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
6F	117.78	17.36	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
7F	117.81	17.48	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
8F	117.62	16.40	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
9F	117.26	16.61	41.08	$e_1 = [29.53, 394.85, 38.67]^T e_{60} = [142.22, 426.98, 43.25]^T$	$e_1 = [44.61522210, 7.924795216, 467.41]^T e_{60} = [44.61551123, 7.926214864, 471.99]^T$	60
10F	117.21	16.91	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
11F	116.97	16.50	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
12F	116.92	16.46	42.15	$e_1 = [28.04, 402.95, 39.70]^T e_{60} = [140.60, 434.30, 44.21]^T$	$e_1 = [44.61529498, 7.924776442, 468.44]^T e_{60} = [44.61557710, 7.926194454, 472.95]^T$	60
13F	116.17	15.74	42.55	$e_1 = [27.85, 405.56, 40.10]^T e_{59} = [139.67, 436.75, 44.63]^T$	$e_1 = [44.61531847, 7.924774048, 468.84]^T e_{59} = [44.61559915, 7.926182737, 473.37]^T$	59
14F	116.25	18.38	42.92	$e_1 = [27.46, 408.01, 40.44]^T e_{58} = [139.28, 439.36, 45.09]^T$	$e_1 = [44.61534052, 7.924769134, 469.18]^T e_{58} = [44.61562263, 7.926177824, 473.83]^T$	58
15F	79.60	17.70	44.18	$e_1 = [62.08, 418.65, 41.32]^T e_{41} = [138.23, 441.72, 45.67]^T$	$e_1 = [44.61543627, 7.925205269, 470.06]^T e_{41} = [44.61564387, 7.926164596, 474.41]^T$	41
16F	78.91	17.56	44.61	$e_1 = [62.21, 421.12, 41.49]^T e_{41} = [137.83, 443.59, 45.82]^T$	$e_1 = [44.61545849, 7.925206906, 470.23]^T e_{41} = [44.61566069, 7.926159556, 474.56]^T$	41

**Fig. 8.** 2D (a) and 3D (b) graphical representations of the algorithm output obtained by processing parcel F (Fig. 1b).

are detailedly reported in [Table 1](#). For indices computation, a set of line class objects was manually drawn by using the QGIS software, each one aligned with a single vine row to be considered as reference detection and localisation of vine rows (ground truth). The procedure, which was performed on a plan view of the 3D point cloud, provided the latitude and longitude coordinates of each manually drawn point ($e_1^*, e_2^*, \dots, e_m^*$)

defining the line object. The altitude of each point was then retrieved by the local digital terrain model (DTM), which was provided by Agisoft Photoscan® software. Please note that the quantity of points representing the location of the r-th vine row provided by the algorithm (n_r) and by the manual procedure (m_r) may differ. As an additional reference, in Parcel F, the position of all the vine rows end points (e_1^* and e_m^*)

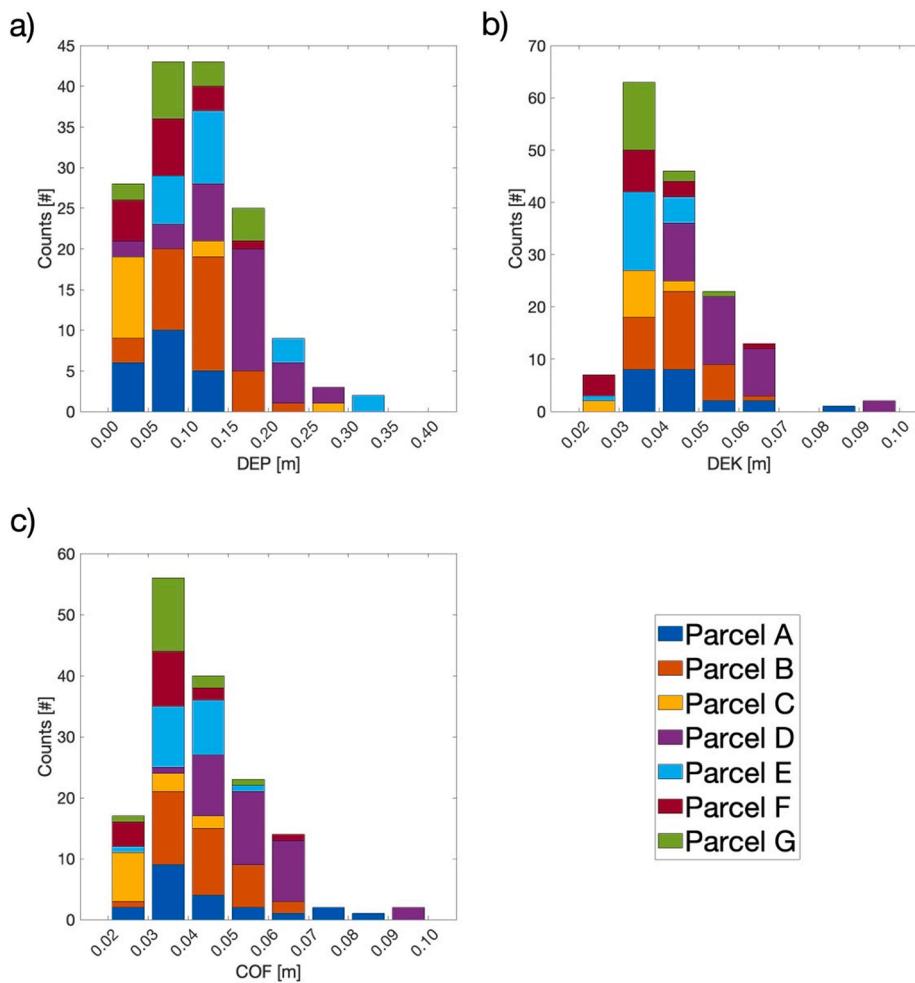


Fig. 9. Histograms of obtained *DEP* (a), *DEK* (b) and *COF* (c) localisation indices of vine row, grouped by parcels. Average and standard deviations of the distributions are reported in Table 3.

were acquired in the field by the S900A Stonex GNSS (Stonex, 2021) with centimetre accuracy.

4. Results and discussion

The detailed results obtained by processing the point cloud of Parcel F (Fig. 1b) are organised in Table 2, while their graphical representation is reported in Fig. 8. Each detected vine row within a parcel is assigned a unique id number, ordered with respect to the southeast-northwest direction. Further outputs of the algorithm are the individual vine row length and average orientation angle, altitude difference, and the spatial coordinates of the points representing the two end points (e_1 and e_n) of the vine row both in the local and WGS84 reference frames. In addition, the table also provides the total number of enhanced key-points E_r used to define the curve γ_r , representing the spatial layout of each vine row. In Appendix A, the algorithm outputs obtained by processing the other six considered parcels are reported for completeness.

Regarding the algorithm performances in detecting the vine rows, the 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 155 vine rows within the seven considered parcels were properly detected and no vine rows were wrongly found. These three performance indices of detection (indices 1, 2 and 3 of Table 1) were computed by comparing the algorithm output with a visual inspection of the considered point clouds and an in-field survey.

Concerning the accuracy of the vine row localisation, 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 (Appendix B). The histograms of the obtained values are reported in Fig. 9 and their average and standard deviation, grouped by processed parcels as well as the overall vine rows, are reported in Table 3. The average *DEP* index of each considered parcel varied between 0.07 (Parcel A) and 0.17 (Parcel D) meters, with standard deviations of 0.04 and 0.14, respectively. Considering the entire processed dataset, the average *DEP* index was 0.12 m with a standard deviation of 0.10 m. The obtained values of *DEP* index proved that the algorithm is able to properly detect the end points of the vine rows in an automatic way from

Table 3
Average and standard deviation of obtained localisation indices of vine row grouped by parcels and, in the last row, of the whole case study.

	<i>DEP</i> [m]		<i>DEK</i> [m]		<i>COF</i> [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 F*	0.28	0.10				
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

* values evaluated using in-field GNSS system measurements.

a vineyard 3D point cloud. Please note that, for Parcel F, the *DEP* index was computed also by considering the points (e_1^* and $e_{m_p}^*$) acquired by the GNNS. In this additional accuracy evaluation, obtained average *DEP* index was 0.28 cm. Such error is mainly due to the difference between the vine row end point definition, which is the canopy edge for the algorithm and the pole trellis location for in field GNNS measurement. Considering the *DEK* index, the obtained average values of each parcel were in the range between 0.04 and 0.06 m with standard deviations of about 0.01. The average *COF* index of the parcels varied between 0.03 and 0.06 m with standard deviations of about 0.01. Both the *DEK* and *COF* indices quantify the error between each vine row location, expressed by the algorithm key points and the manual ones. The 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 a few centimetres, show that the algorithm outputs, in term of vine row location, are compatible with the requirements of precision agriculture operations, such as UGV path planning and autonomous guidance (Donati et al., 2021).

All the data processing has been performed on a 2.9 GHz 6-Core Intel Core i9, with 32 GB DDR4 RAM memory of 2400 MHz. The average computing time for processing a point cloud with a density of about 1500 points/m² and representing a 1 ha vineyard was about 90 min.

5. Conclusions

In this work, an innovative 3D point cloud processing algorithm for vine row detection and localisation within a vineyard map, based on the detection of key points and a density-based clustering approach, was presented. The algorithm provides as outputs, in a fully automated way, an information set which fully describes the spatial location of each vine row, even when not straight, within the whole 3D model of the agricultural environment. This is a crucial phase in the interpretation of the complex and huge 3D point clouds of agricultural environments, focusing from a macro level (parcel and plot scale) to a micro level (plants, fruits, branches).

The algorithm does not require colour or specific spectral vine information and is therefore independent of the adopted airborne sensor used to acquire the in-field data or imagery. In addition, the proposed approach proved to be robust to any occurrence of missing plants and any vine row layout and orientation, including curvilinear or not straight rows.

The unsupervised detection of each single vine row within a whole vineyard model is essential not only for enabling the automatic application of further specific processing algorithms at plant scale, but also for in-field path planning purposes. 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 UGV operations in the field (Mammarella et al., 2022b). Indeed, by overcoming 2D path planning algorithms limits, 3D path planning fully exploits the terrain's and environmental characteristics.

Funding

This research was partially funded by the projects “*Marcel – Macchine Agricole Elettriche*” (POR FESR 2014–2020 – Piattaforma Bioeconomia – Piedmont Region - Codice domanda: 333-190) and ““*DATA-BUS - Digital Agriculture Technology to Achieve data to Build User-friendly Sustainability indicators*” (PRIN 2020, Prot. 2020SCNF4L).

CRediT authorship contribution statement

A. Biglia: Data curation, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **S. Zaman:** Data curation, Validation, Formal analysis, Investigation, Writing – original draft. **P. Gay:** Conceptualization, Funding acquisition, Project

administration, Writing – review & editing. **D. Ricauda Aimonino:** Funding acquisition, Methodology, Writing – review & editing. **L. Comba:** Conceptualization, Investigation, Funding acquisition, Methodology, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the following companies: Azienda Agricola Germano Ettore (Serralunga d'Alba, Italy) and Azienda Agricola G. D. Vajra (Barolo, Italy) for hosting the experiments.

Appendices A & B. Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.compag.2022.107166>.

References

- Bechar, A., Vigneault, C., 2017. Agricultural robots for field operations. Part 2: Operations and systems. Biosyst. Eng. 153, 110–128. <https://doi.org/10.1016/j.biosystemseng.2016.11.004>.
- Biosca, J.M., Lerma, J.L., 2008. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS J. Photogramm. Remote Sens. 63, 84–98. <https://doi.org/10.1016/j.isprsjprs.2007.07.010>.
- Blanquart, J.-E., Sirignano, E., Lenaerts, B., Saeys, W., 2020. Online crop height and density estimation in grain fields using LiDAR. Biosyst. Eng. 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, I.R., Sankaran, S., Jacoby, P.W., 2019. Evaluation of mobile 3D light detection and ranging based canopy mapping system for tree fruit crops. Comput. Electron. Agr. 158, 284–293. <https://doi.org/10.1016/j.compag.2019.02.012>.
- Chen, H., Liang, M., Liu, W., Wang, W., Liu, P.X., 2022. An approach to boundary detection for 3D point clouds based on DBSCAN clustering. Pattern Recogn. 124, 108431. <https://doi.org/10.1016/j.patcog.2021.108431>.
- 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. Comput. Electron. Agr. 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 Sens. 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>.
- 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., Zaman, S., Biglia, A., Ricauda Aimonino, D., Dabbene, F., Gay, P., 2020a. 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>.
- Comba, L., Biglia, A., Ricauda Aimonino, D., Tortia, C., Mania, E., Guidoni, S., et al., 2020b. 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., Biglia, A., Ricauda Aimonino, D., Barge, P., Tortia, C., Gay, P., 2021a. Semantic interpretation of multispectral maps for precision agriculture: a machine learning approach. Precis. Agric. 21, 291–297. https://doi.org/10.3920/978-90-8686-916-9_34.
- Comba, L., Biglia, A., Ricauda Aimonino, D., Barge, P., Tortia, C., Gay, P., 2021b. Neural network clustering for crops thermal mapping. Acta Hortic. 1311, 513–520. <https://doi.org/10.17660/ActaHortic.2021.1311.65>.
- 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>.

- D'Errico, J., 2021. Interparc (<https://www.mathworks.com/matlabcentral/fileexchange/34874-interparc>), MATLAB Central File Exchange. Retrieved October 8, 2021.
- De Boor, C., 1978. A Practical Guide to Splines. Springer-Verlag, New York.
- Díaz, C.A., Sebastián Pérez, D., Miatello, H., Bromberg, F., 2018. Grapevine buds detection and localization in 3D space based on Structure from Motion and 2D image classification. Comput. Ind. 99, 303–312. <https://doi.org/10.1016/j.compind.2018.03.033>.
- Donati, C., Mammarella, M., Comba, L., Biglia, A., Dabbene, F., Gay, P., 2021. Improving agricultural drone localization using georeferenced low-complexity maps, 2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 372–377. <https://doi.org/10.1109/MetroAgriFor52389.2021.9628607>.
- Dusadeerungskul, P.O., Nof, S.Y., 2019. A collaborative control protocol for agricultural robot routing with online adaptation. Comput. Ind. Eng. 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., et al., 2017. Mobile terrestrial laser scanner applications in precision fruticulture/horticulture and tools to extract information from canopy point clouds. Precis. Agric. 18, 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. In: Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining, pp. 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, 105412. <https://doi.org/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. Comput. Electron. Agr. 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. Biosyst. Eng. 153, 28–41. <https://doi.org/10.1016/j.biosystemseng.2016.10.019>.
- Grella, M., Marucco, P., Zwertyaeghe, I., Gioelli, F., Bozzer, C., Biglia, A., et al., 2022. The effect of fan setting, air-conveyor orientation and nozzle configuration on airblast sprayer efficiency: Insights relevant to trellised vineyards. Crop Prot. 155, 10921. <https://doi.org/10.1016/j.cropro.2022.105921>.
- Guidoni, S., Drory, E., Comba, L., Biglia, A., Ricauda Aimonino, D., Gay, P., 2021. A method for crop water status evaluation by thermal imagery for precision viticulture: preliminary results. Acta Hortic. 1314, 83–90. <https://doi.org/10.17660/ActaHortic.2021.1314.12>.
- 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>.
- Kerkchek, 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., Nikbakht, E., 2020. Field path planning using capacitated arc routing problem. Comput. Electron. Agr. 173, 105401. <https://doi.org/10.1016/j.compag.2020.105401>.
- Kong, Q., Siauw, T., Bayen, A.M., 2021. Chapter 17 – Interpolation. Python Programming and Numerical Methods. Academic Press, pp. 295–313 <https://doi.org/10.1016/B978-0-12-819549-9.00027-0>.
- Li, J., Cheng, X., Xiao, Z., 2022. A branch-trunk-constrained hierarchical clustering method for street trees individual extraction from mobile laser scanning point clouds. Measurement 189, 110440. <https://doi.org/10.1016/j.measurement.2021.110440>.
- 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. Comput. Electron. Agr. 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, 2177–2194. <https://doi.org/10.3390/s110202177>.
- Malavazi, F.B.P., Guyonneau, R., Fasquel, J.-B., Lagrange, S., Mercier, F., 2018. LiDAR-only based navigation algorithm for an autonomous agricultural robot. Comput. Electron. Agr. 154, 71–79. <https://doi.org/10.1016/j.compag.2018.08.034>.
- Mammarella, M., Comba, L., Biglia, A., Dabbene, F., Gay, P., 2020. Cooperative agricultural operations of aerial and ground unmanned vehicles. IEEE International Workshop on Metrology for Agriculture and Forestry, 224–229.
- Mammarella, M., Comba, L., Biglia, A., Dabbene, F., Gay, P., 2022a. Cooperation of unmanned systems for agricultural applications: A theoretical framework. Biosyst. Eng. In press. <https://doi.org/10.1016/j.biosystemseng.2021.11.008>.
- Mammarella, M., Comba, L., Biglia, A., Dabbene, F., Gay, P., 2022b. Cooperation of unmanned systems for agricultural applications: A case study in a vineyard. Biosyst. Eng. In press. <https://doi.org/10.1016/j.biosystemseng.2021.12.010>.
- 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, 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. Comput. Electron. Agr. 163, 104827. <https://doi.org/10.1016/j.compag.2019.05.036>.
- Michels, M., von Hobre, C.-F., Musshoff, O., 2020. A trans-theoretical model for the adoption of drones by large-scale German farmers. J. Rural Stud. 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>.
- Nagasaki, Y., Saito, H., Tamaki, K., Seki, M., Kobayashi, K., Taniwaki, K., 2009. An autonomous rice planter guided by global positioning system and inertial measurement unit. J. Field Rob. 26, 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. Comput. Electron. Agr. 178, 105702. <https://doi.org/10.1016/j.compag.2020.105702>.
- 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. Eur. J. Rem Sens. 50, 179–186. <https://doi.org/10.1080/2297254.2017.1308234>.
- Sanz, R., Llorens, J., Escolà, A., Arnó, J., Planas, S., Román, C., et al., 2018. LIDAR and non-LIDAR-based canopy parameters to estimate the leaf area in fruit trees and vineyard. Agr. Forest Meteorol. 260–261, 229–239. <https://doi.org/10.1016/j.agrformet.2018.06.017>.
- Stonex, 2021 from <https://www.stonex.it/project/s900a-new-gnss-receiver/>.
- Sun, S., Li, C., Chee, P.W., Paterson, A.H., Jiang, Y., Xu, R., et al., 2020. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. ISPRS J. Photogramm. 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 Sens. 12, 1597. <https://doi.org/10.3390/rs12101597>.
- Wang, 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. Comput. Electron. Agr. 175, 105523. <https://doi.org/10.1016/j.compag.2020.105523>.
- Weinmann, M., Jutzl, B., Hinz, S., Mallett, 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, M.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. Comput. Electron. Agr. 179, 105837. <https://doi.org/10.1016/j.compag.2020.105837>.
- Zeng, L., Feng, J., He, L., et al., 2020. Semantic segmentation of sparse 3D point cloud based on geometrical features for trellis-structured apple orchard. Biosyst. Eng. 196, 46–55. <https://doi.org/10.1016/j.biosystemseng.2020.05.015>.
- Zhang, C., Zhang, K., Ge, L., Zou, K., Wang, S., Zhang, J., Li, W., 2021. A method for organ classification and fruit counting on pomegranate trees based on multi-features fusion and support vector machine by 3D point cloud. Sci. Hortic. 278, 109791. <https://doi.org/10.1016/j.scientia.2020.109791>.
- 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. Comput. Electron. Agr. 162, 143–153. <https://doi.org/10.1016/j.compag.2019.04.014>.