Research Paper

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

Lorenzo Comba [a,b,*], Shahzad Zaman [a], Alessandro Biglia [a], Aimonino Davide Ricauda [a], Fabrizio Dabbene [b], Paolo Gay [a]

[a] Department of Agricultural, Forest and Food Sciences (DiSAFA) — Università degli Studi di Torino, Largo Paolo Braccini 2, 10095, Grugliasco (TO), Italy
[b] CNR-IEIIT — Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

ARTICLE INFO

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 scenarios 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. 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 amount 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.

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 (Comba et al., 2020; Khaliq et al., 2019;

Nomenclature

a	Dimensions of vine row section $\mathcal{S}_k$ along x_k axis [m] (model parameter)
b	Distance between two sequential vine row sections $\mathcal{S}_k$ and $\mathcal{S}_{k+1}$ [m] (model parameter)
c	Number of vertices of c-gon $\mathcal{P}^c$ (model parameter)
R_k	*Complexity reduction* index of 3D mesh $\mathcal{M}$
$\mathcal{C}_k$	Set of points representing the kth canopy section ($\mathcal{C}_k = \mathcal{C}_k^- \cup \mathcal{C}_k^+$)
$\mathcal{C}^{2D}$	Two-dimensional projection of $\mathcal{C}$ on the plane x = 0
$\tilde{\mathcal{C}}^{2D}$	Outlier-filtered $\mathcal{C}^{2D}$ set of points
$F_{k,k+1}$	Set of triangular faces of the generated model mesh, between vertices V_k and V_{k+1}
G_k	*Good-modelling* index of 3D mesh $\mathcal{M}$
h_k	Peak location of H_z
$H_y(\mathcal{S}_k, s)$	Normalised frequencies distribution histogram of points $p_i \in \mathcal{S}_k$ along y_k axes
$H_z(\mathcal{S}_k, t)$	Normalised frequencies distribution histogram of points $p_i \in \mathcal{S}_k$ along z_k axes
$\mathcal{H}$	Convex hull of points set $\mathcal{C}^{2D}$
$\mathcal{K}$	Set of all the considered vine row section $\mathcal{S}_k$
L_k'	and L_k'' Lines defining plane $\wp_k$
$\mathcal{M}$	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 $\mathcal{M}$
$O_{\text{LOC}_k}^{\{WGS84\}}$	Origin of local reference frame LOC_k in WGS84 coordinates
$\mathcal{P}^c$	c-gon containing the point set $\tilde{\mathcal{C}}^{2D}$ with vertices
$\mathcal{P}[V^{c*}]$	Minimum area c-gon containing the point set $\tilde{\mathcal{C}}^{2D}$
$\mathcal{PC}$	3D point cloud of vineyard
$\wp_k^+$	Plane defined by two lines L_k' and L_k''
$\mathcal{Q}_k$	Model $\mathcal{M}_k$ quality score
s_y	Bin of the histogram H_y
s_z	Bin of the histogram H_z
$\mathcal{S}_k$	Subset of points representing a section of vine row
$\mathcal{S}_k^+$ and $\mathcal{S}_k^-$	Two sides of vine row section $\mathcal{S}_k$ with $y \geq 0$ and $y < 0$ respectively

u_i	ith 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
v_i	ith vertex of c-gon $\mathcal{P}^c$ in the 2D plane x = 0
v_i	ith 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}	$\mathcal{C}_k$ 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 {LOC}, tangent to the local wine row direction ϑ_k
y_k	y axis of the {LOC}
$y_{k,max}$	Greatest value of y coordinates of points in $\mathcal{S}_k$
Y_k	Bins set of the histogram H_y
z_k	z vertical axis of the {LOC}
$z_{k,max}$	Greatest value of z coordinates of points in $\mathcal{S}_k$
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
{LOC}	Local metrical reference frame
{WGS84}	World geodetic system 1984

Reza, Na, Baek, & Lee, 2019; Sozzi, Kayad, Giora, Sartori, & Marinello, 2019; Mazzia, Comba, Khaliq, Chiaberge, & Gay, 2020), whilst minimising the use of resources (Higgins, Schellberg, & Bailey, 2019; Peng et al., 2019) and improving environmental protection (Grella, Gallart, Marucco, Balsari, & Gil, 2017; Oberti et al., 2016). 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 (Comba et al., 2019; Grimstad & From, 2017; Primicerio et al., 2017; Utstumo et al., 2018), 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, Ricauda Aimonino, & Gay, 2016; Reina, Milella, & Galati, 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 (Bechar & Vigneault, 2016; Vidoni, Bietresato, Gasparetto, & Mazzetto, 2015). 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;

Primicerio et al., 2015; Van Henten, Bac, Hemming, & Edan, 2013; 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, la Cour-Harbo, & Osen, 2016). This requires the development of new 3D models, such as point clouds or triangulated meshes (Miranda-Fuentes, Llorens, Gamarra-Diezma, Gil-Ribes, & Gil, 2015; Weiss & Biber, 2011). 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, Lenz, Teutrine, & Steinhage, 2017) and by a family of devices known as depth cameras (Condotta, Brown-Brandl, Pitla, Stinn, & Silva-Miranda, 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 (Bao, Tang, Srinivasan, & Schnable, 2019; Rosell-Polo et al., 2017). 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, Zhou, Vories, Sudduth, & Zhang, 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, Moeckel, Hensgen, & Wachendorf, 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, Biglia, Ricauda Aimonino, & Gay, 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 (Pavón-Pulido, López-Riquelme, Torres, Morais, & Pastor, 2017; Van Evert et al., 2017; Wolfert, Ge, Verdouw, & Bogaardt, 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).

This paper presents 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 paper is structured as follows: section 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 3, while section 4 reports the conclusions and future developments. To better visualise the 3D models, a set of animations are available online as supplementary material (mp4 video files) in the electronic version of the article.

Supplementary video related to this article can be found at https://doi.org/10.1016/j.biosystemseng.2020.05.013

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. 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

$$\mathcal{PC}^{\{WGS84\}} = \left\{ [\phi_i, \ \lambda_i, \ e_i]^T \in \mathbb{R}^3; \ i = 1, \ ..., \ N_{\mathcal{PC}} \right\}, \tag{1}$$

where ϕ_i, λ_i and e_i are, respectively, the latitude, longitude and elevation coordinates of the ith 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 1000 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
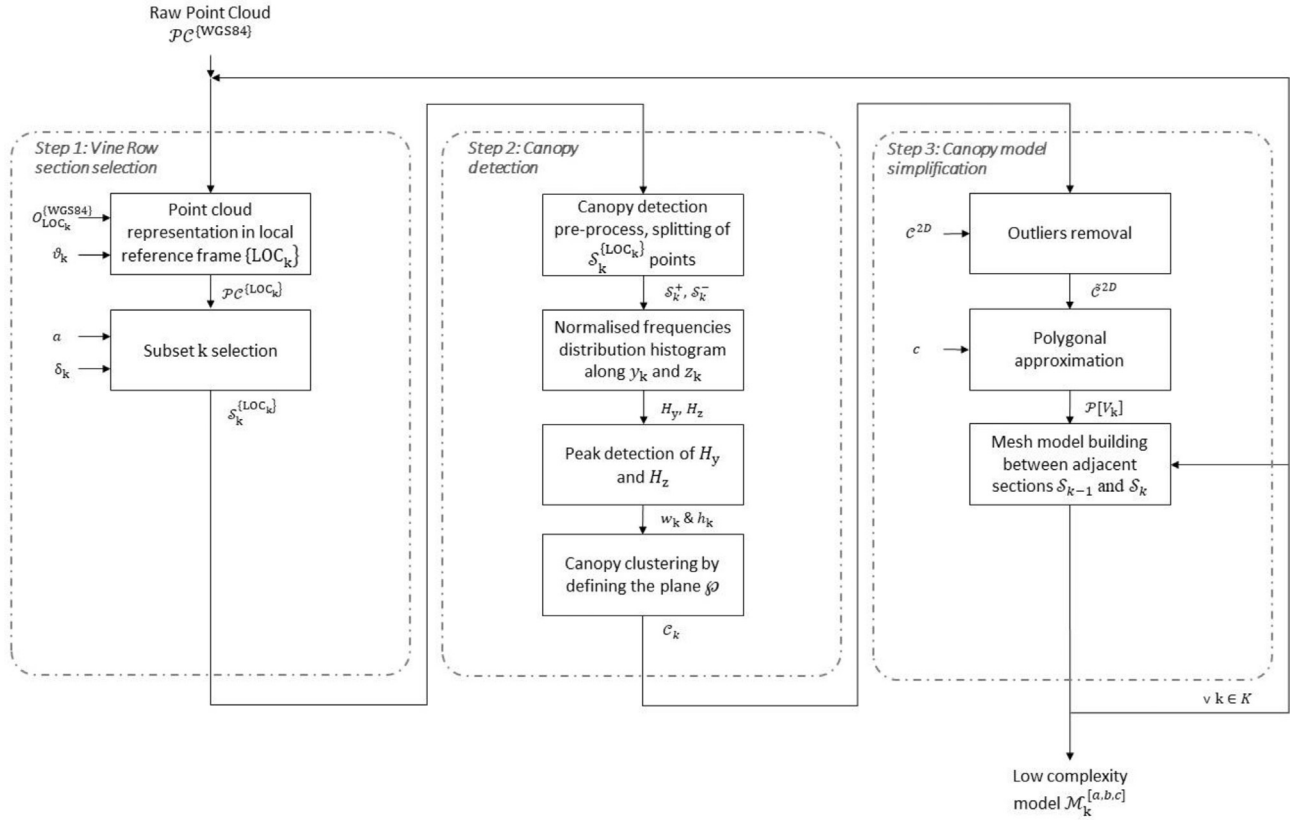
Fig. 1 – Scheme of the defined modelling framework to generate low complexity 3D mesh models of vine rows from raw 3D point clouds.

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.

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 $\mathcal{S}_k$ representing a vine row section from the whole $\mathcal{PC}^{\{WGS84\}}$ (Fig. 2). This process was performed by defining a local metrical reference frame $\{LOC_k\}$ 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). The origin of $\{LOC_k\}$ was defined in $\mathbf{O}_{LOC_k}^{\{WGS84\}} \in \gamma$, so that the distance along the vine row centre line γ between two local reference systems $\{LOC_{k-1}\}$ and $\{LOC_k\}$, and thus between two vineyard subsets $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$, is equal to b, satisfying the line integral

$$\int_{t_{k-1}}^{t_k} \gamma'(t)\,dt = b \qquad (2)$$

where $\gamma(t_{k-1}) = \mathbf{O}_{LOC_{k-1}}$ and $\gamma(t_k) = \mathbf{O}_{LOC_k}$. The x_k axis of $\{LOC_k\}$ 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).

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

$$\mathcal{S}_k^{\{LOC_k\}} = \left\{ [x, y, z]^T \in \mathcal{PC}^{\{LOC_k\}} \,\middle|\, |x| \leq \frac{a}{2}, \; |y| \leq \frac{\delta_k}{2} \right\} \qquad (3)$$

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 $card(\mathcal{S}_k^{\{LOC_k\}})$ (lower limit), and allow the vine-row section to be considered as rectilinear (upper limit). A sample of subset $\mathcal{S}_{268}$ centred in $\mathbf{O}_{LOC_{268}}^{\{WGS84\}} = [44.62447°\; 8.00105°\; 364.6\text{ m}]^T$ is shown in Figs. 2 and 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 $\{LOC_k\}$ will be omitted.

2.2. Semantic interpretation for vine canopy detection

Once subset $\mathcal{S}_k$ is selected, the next step consists in automatically detecting the set of points $\mathcal{C}_k$ representing the canopy, distinguishing it from those representing the inter-row
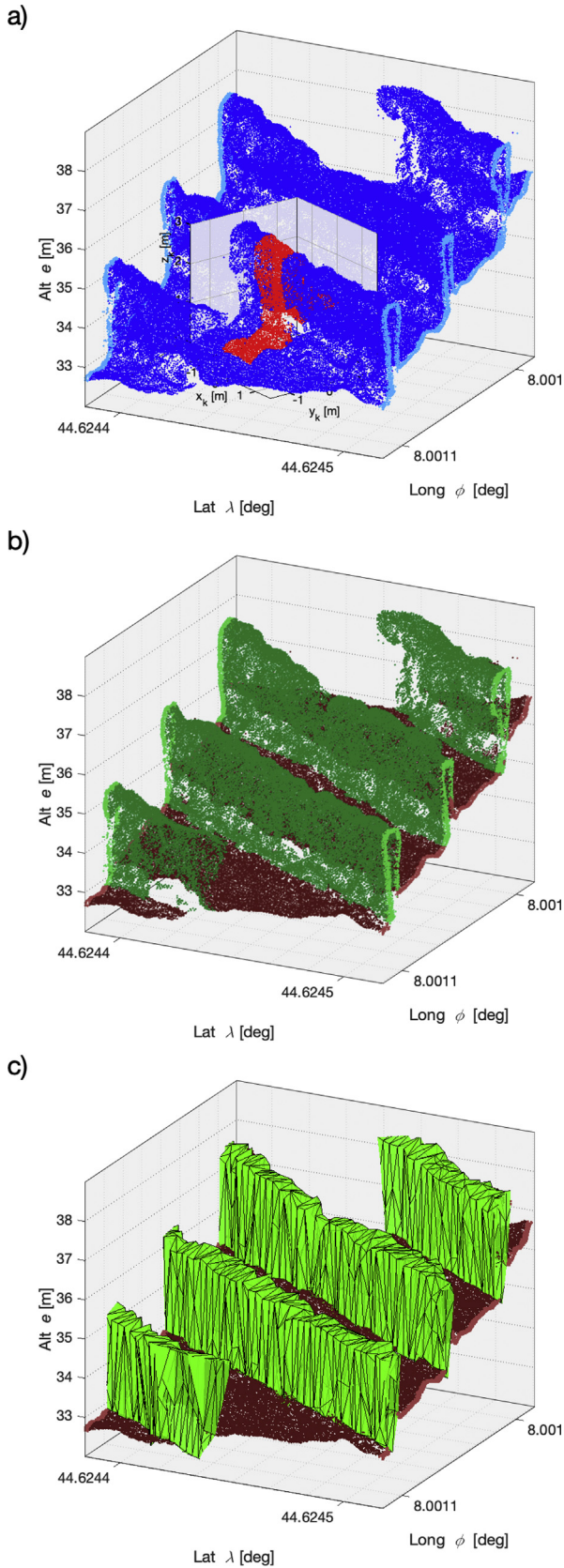
terrain. Since the terrain elevation of two adjacent inter rows may differ in vineyards located in hilly regions, the classification is performed by individually considering each side of the vine row $\mathcal{S}_k^+$ and $\mathcal{S}_k^-$ (Fig. 4). Being the origin of the reference system $\{LOC_k\}$ located in the centre line of the canopy width, $\mathcal{S}_k^+$ and $\mathcal{S}_k^-$ can easily be defined as

$$\mathcal{S}_k^+ = \left\{ [x, y, z]^T \in \mathcal{S}_k \,|\, y \geq 0 \right\} \tag{4}$$

and

$$\mathcal{S}_k^- = \left\{ [x, y, z]^T \in \mathcal{S}_k \,|\, y < 0 \right\} \tag{5}$$

Focusing on side $\mathcal{S}_k^+$ of wine row section $\mathcal{S}_k$, the classification was obtained by determining a plane $\wp_k^+$ representing the boundary of the two regions containing respectively the points representing the terrain and those representing the canopy, (Fig. 4f). Plane $\wp_k^+$ was defined as the plane passing through the two lines parallel to the x_k axis

$$L_k' = \left\{ [x, y, z]^T \in \mathbb{R}^3 \,|\, y = 0, \ z = 0 \right\} \tag{6}$$

and

$$L_k'' = \left\{ [x, y, z]^T \in \mathbb{R}^3 \,|\, y = w_k^+, \quad z = h_k^+ \right\} \tag{7}$$

where w_k is related to the location along the y_k axis of the external surface of the canopy wall and h_k is the inter-row path terrain elevation along the z_k axis (Fig. 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

$$H_y\big(\mathcal{S}_k^+, \, s\big) = \text{card}\left\{ p_i = [x, y, z]^T \in \mathcal{S}_k^+ : |y - s_y| < \frac{\delta_s}{2} \right\} \cdot \text{card}\big(\mathcal{S}_k^+\big)^{-1} \tag{8}$$

where $s_y \in Y_k = \{0, \ \delta_s, 2\delta_s, \ldots, y_{k,max}\}$, Y_k is the set of all the histogram bins, δ_s is the bin width and $y_{k,max}$ is the highest value of the considered y coordinates (Fig. 4e). Analogously, the value of h_k is the peak of the normalised frequencies distribution histogram

$$H_z\big(\mathcal{S}_k^+, t\big) = \text{card}\left\{ p_i = [x, y, z]^T \in \mathcal{S}_k^+ : |z - s_z| < \frac{\delta_s}{2} \right\} \cdot \text{card}\big(\mathcal{S}_k^+\big)^{-1} \tag{9}$$

where $s_z \in Z_k = \{0, \ \delta_s, 2\delta_s, \ldots, z_{k,max}\}$, Z_k is the set of all the histogram bins and $z_{k,max}$ is the highest value of the considered z coordinates (Fig. 4d). In Fig. 4f, plane $\wp_{268}^+$, defined by line L_{268}'' with $w_{268} = 0.22$ and $h_{268} = 0.86$, is displayed. Point cloud subset $\mathcal{C}_k^+$ representing the canopy wall of the considered side of vine row section $\mathcal{S}_k^+$ can be thus determined as

$$\mathcal{C}_k^+ = \left\{ [x, y, z]^T \in \mathcal{S}_k^+ \,\left|\, z \geq \frac{h_k}{w_k} y \right. \right\}. \tag{10}$$

Performing this procedure to both subsets $\mathcal{S}_k^+$ and $\mathcal{S}_k^-$, the set of all points representing canopy $\mathcal{C}_k$ for the kth section can be obtained by the union of sets $\mathcal{C}_k^+$ and $\mathcal{C}_k^-$, that is $\mathcal{C}_k = \mathcal{C}_k^- \cup \mathcal{C}_k^+$,

Fig. 2 – (a) Portion of the raw 3D point cloud $\mathcal{PC}^{\{WGS84\}}$ (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). Animation of this Figure can be find in the additional material in the electronic version of this manuscript.
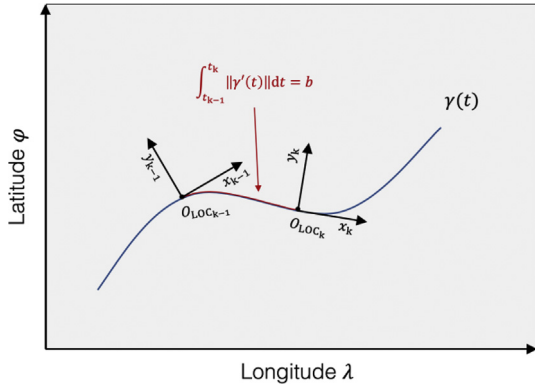
Fig. 3 – Local reference systems $\{LOC_{k-1}\}$ and $\{LOC_k\}$ for vineyard subsets $\mathcal{S}_{k-1}$ and $\mathcal{S}_k$ definition, with origin in $\gamma(t_{k-1}) = O_{LOC_{k-1}}$ and $\gamma(t_k) = O_{LOC_k}$, respectively.

with $N_{\mathcal{C}_k} = \text{card}(\mathcal{C}_k)$. The results of this clustering procedure for canopy detection, obtained by processing sample subset $\mathcal{S}_{268}^{+}$ and the whole point cloud $\mathcal{PC}^{\{WGS84\}}$, are shown in Fig. 4f and Fig. 2b, respectively.

2.3. Canopy model simplification

In this section, the processing step aimed at reducing the complexity (and density) of point set $\mathcal{C}_k$ 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 kth 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 $x = 0$

$$\mathcal{C}^{2D} = \left\{ [x,y]^T \in \mathbb{R}^2 \,\middle|\, x = y, \ y = z, [x,y,z]^T \in \mathcal{C} \right\} \tag{11}$$

A graphical representation of set $\mathcal{C}_{268}^{2D}$ relative to section $k = 268$ is shown in Fig. 5a.

2.3.1. Outlier removal

Examining Fig. 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, pp. 2–8).

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

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

where $U^{2D} = \{u_i = [\zeta_i, \eta_i]^T, i = 1, ..., N_{\mathcal{H}}\}$ are the vertices of the polygon. Recall that, by construction, the vertices of $\mathcal{H}[U^{2D}]$ represent a subset of the points in $\mathcal{C}^{2D}$.

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^{2D} is now reported. This formula is termed *Gauss's area formula* or *shoelace formula*, see Boland and Urrutia (2000, pp. 159–162).

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

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

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^{2D} determined (these are also points in $\mathcal{C}^{2D}$); 2) the vertices of $\mathcal{H}[U^{2D}]$ were removed one-by-one from $\mathcal{C}^{2D}$, 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)

Input: 2D point set. $\mathcal{C}^{2D}$

Output: an outlier-filtered 2D point set. $\tilde{\mathcal{C}}^{2D}$

0. Let $j = 0$ and set $\mathcal{C}^{[0]} = \mathcal{C}^{2D}$
1. Compute $\mathcal{H}[U^{[j]}] = convhull(\mathcal{C}^{[j]})$
2. For $\ell = 1$ to $\text{card}(U^{[j]})$
   a. Compute $\mathcal{P}[U_{\searrow\ell}^{[j]}] = convhull(\mathcal{C}_{\searrow\ell}^{[j]})$, with $\mathcal{C}_{\searrow\ell}^{[j]} = \mathcal{C}^{[j]} \setminus \{v_\ell\}$
3. Let $\mathcal{C}^{[j+1]} = \mathcal{C}_{\searrow\ell^*}^{[j]}$ with $\ell^* = \underset{\ell}{\arg\min} Area(\mathcal{H}[U_{\searrow\ell}^{[j]}])$
4. If EXITCOND return $\tilde{\mathcal{C}}^{2D} = \mathcal{C}^{[j+1]}$, else let $j = j + 1$ and go to 1.
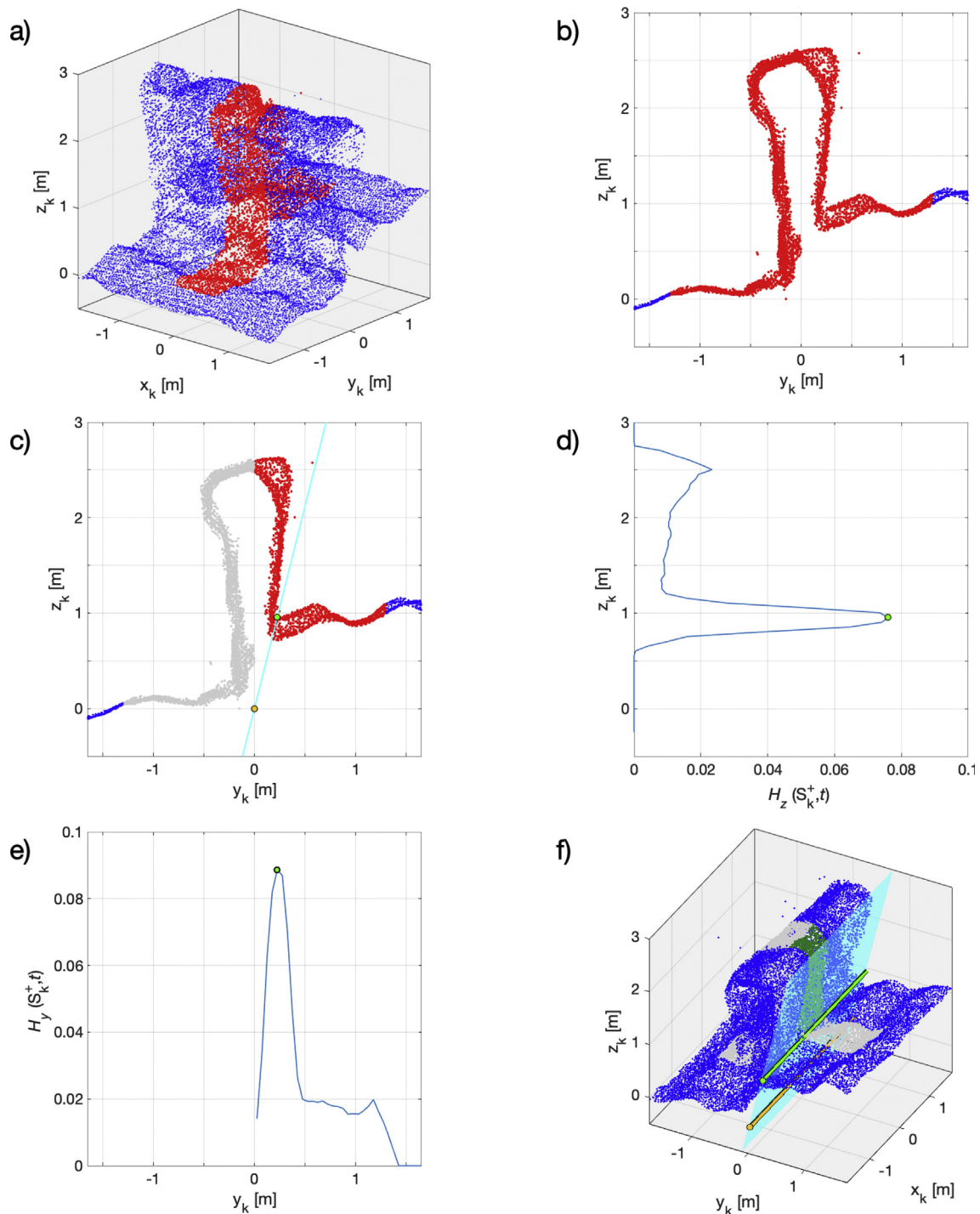
Fig. 4 − **(a) 3D view and (b) 2D view of the sample subset $\mathcal{S}_{268}^{\{LOC_k\}}$ (red dots) of $\mathcal{PC}^{\{WGS84\}}$ (blue dots), located in $O_{LOC_{268}}^{\{WGS84\}} =$ [44.62447° 8.00105° 364.6 m] and defined with $\vartheta_{268} = 63.4°$, $\delta_{268} = 2.6$ m and $a = 0.8$; (c) 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 $H_z(\mathcal{S}_{268}^{+}, s)$ and peak location h_k (green); (e) normalised frequencies distribution histogram $H_y(\mathcal{S}_{268}^{+}, s)$ (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). Animation of this Figure can be find in the additional material in the electronic version of this manuscript.**
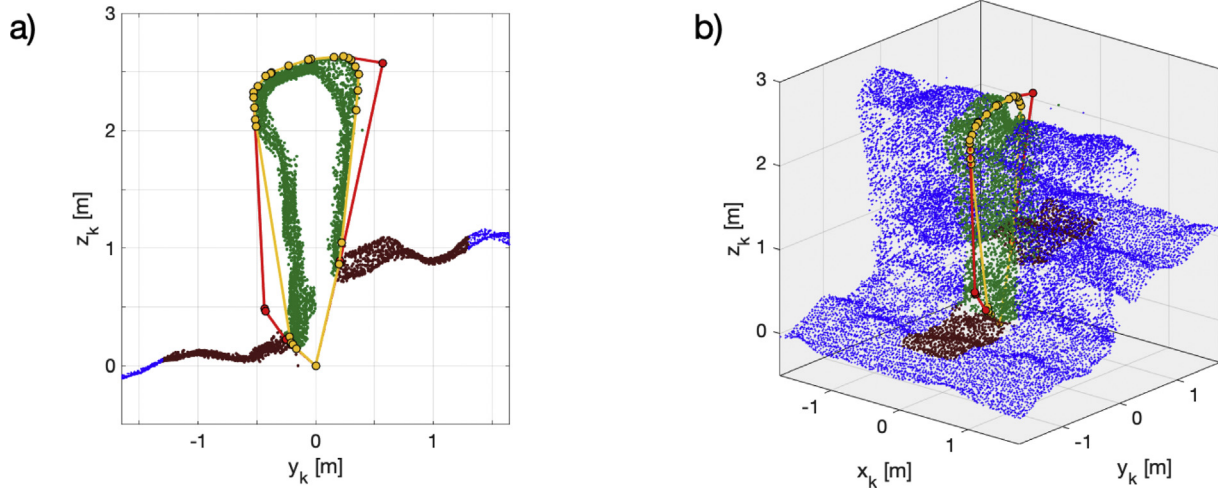
Fig. 5 – (a) Convex hull polygon $\mathcal{H}[U^{[0]}]$ (red) enclosing all $\mathcal{C}^{2D}_{268}$ points (green dots) and convex hull polygon $\mathcal{H}[U^{[8]}]$, after removing 8 outliers (orange); and (b) their 3D view. Animation of this Figure can be find in the additional material in the electronic version of this manuscript.

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 $\Delta A^{[j]} = Area(\mathcal{C}^{[j]}_{\searrow \varrho^*}) - Area(\mathcal{C}^{[j-1]}_{\searrow \varrho^*})$). 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 $\mathcal{C}^{2D}$, since at each step it requires the computation of card($U^{[j]}$) convex hulls. The worst possible case is when all points of $\mathcal{C}^{2D}$ belong to the convex hull (e.g. point on a circumference): in this case, the complexity of removing one outlier is of the order $O(N_C^2 \log N_C)$. Some steps of the procedure of outlier removal are shown in Fig. 5, where a set of convex hull $\mathcal{H}[U]$ are shown for the processing of $\mathcal{C}^{2D}_{268}$.

2.3.2. Polygonal approximation

To approximate the outlier-filtered 2D point set $\tilde{\mathcal{C}}^{2D}$, 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 $\mathcal{P}^c = \mathcal{P}[V^{2D}]$ is defined as a two dimensional polytope (polygon) with c vertices

$$V^{2D} = \{ \boldsymbol{v}_i = [\zeta_i, \eta_i]^T, i = 1, ..., c \} \tag{14}$$

The vertices are assumed to be *ordered in a counter-clockwise way*. An example of c-gon $\mathcal{P}^c = \mathcal{P}[V^{2D}]$ is given in Fig. 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 $\mathcal{C}^{2D} = \{ \boldsymbol{p}_i = [x_i, y_i]^T \in \mathbb{R}^2, i = 1, ..., N_C \}$, find the c-gon $\mathcal{P}^c = \mathcal{P}[V^{2D}]$ of minimum area such that $\mathcal{C}^{2D} \subseteq \mathcal{P}^c$. This is formulated as follows:

$$\mathcal{P}[V^{2D*}] = \underset{V^{2D}}{\arg\min} \, Area(\mathcal{P}[V^{2D}])$$
$$\text{s.t.} \quad \boldsymbol{p}_i \in \mathcal{P}[V^{2D}], \, i = 1, ..., N_C \tag{15}$$

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

$$(\boldsymbol{\zeta}^*, \boldsymbol{\eta}^*) = \underset{(\zeta, \eta)}{\arg\min} \boldsymbol{\zeta}^T \mathbf{S} \boldsymbol{\eta}$$
$$\text{s.t.} \; [\zeta_j \zeta_{j+1}] \mathbf{D} \begin{bmatrix} \eta_j \\ \eta_{j+1} \end{bmatrix} + \boldsymbol{d}^T y_i \begin{bmatrix} \zeta_j \\ \zeta_{j+1} \end{bmatrix} - \boldsymbol{d}^T x_i \begin{bmatrix} \eta_j \\ \eta_{j+1} \end{bmatrix} \leq 0,$$
$$j = 1, ..., c - 1 \quad i = 1, ..., N_C$$
$$[\zeta_c \zeta_1] \mathbf{D} \begin{bmatrix} \eta_c \\ \eta_1 \end{bmatrix} + \boldsymbol{d}^T y_i \begin{bmatrix} \zeta_c \\ \zeta_1 \end{bmatrix} - \boldsymbol{d}^T x_i \begin{bmatrix} \eta_c \\ \eta_1 \end{bmatrix} \leq 0,$$
$$i = 1, ..., N_C \tag{16}$$

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

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

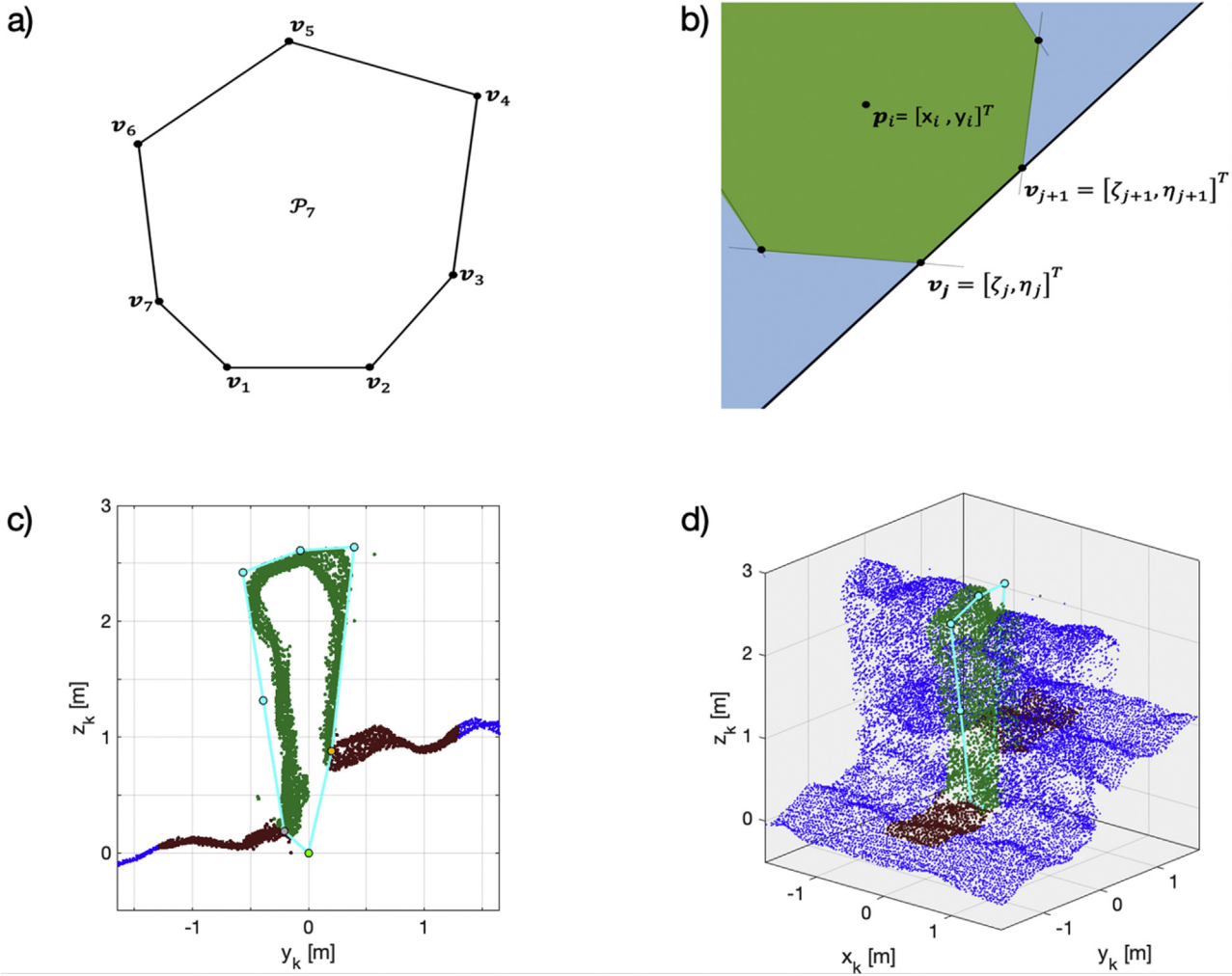Before sharing proof of the above result, a few

Fig. 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 $\mathcal{C}^{2D}_{268}$ (cyan line) with vertices V^{2D}_{268} (of which $v_1 = [0\ 0]$ (green dot), $v_2 = [w^+_{268} h^+_{268}] = [0.22\ 0.87]$ (orange dot) and $v_7 = [w^-_{268} h^-_{268}] = [-0.28\ 0.26]$ (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 $V^{\{LOC_{268}\}}_{268}$. Animation of this Figure can be find in the additional material in the electronic version of this manuscript.

considerations should be made: first, that Eq. (16) was indeed noted to be a bilinear problem, as cost $\zeta^T S \eta$ is bilinear (note that matrix S is skew-symmetric by construction), and also that the constraints are bilinear equations of varsiables (ζ, η).

Proof of Theorem 1.

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

$$Area(\mathcal{P}[V^c])$$

$$= \frac{1}{2}\left[\left(\sum_{i=1}^{c-1} \zeta_i \eta_{i+1} + \zeta_n \eta_i\right) - \left(\sum_{i=1}^{c-1} \zeta_{i+1} \eta_i + \zeta_i \eta_n\right)\right]$$

$$= \frac{1}{2}\left[(\zeta^T \tilde{S} \eta) - (\eta^T \tilde{S} \zeta)\right]$$

$$= \frac{1}{2}\left[(\zeta^T \tilde{S} \eta) - (\zeta^T \tilde{S}^T \eta)\right] = \frac{1}{2}\zeta^T S \eta \quad (18)$$

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 p_i is contained in the c-gon $\mathcal{P}[V^{2D}]$, it must lie on the left of the vector $(v_{j+1} - v_j)$, for all j (see Fig. 6b). This is equivalent to imposing that the sign of the cross (external) product of vector $(v_{j+1} - v_j)$ with vector $(p_i - v_j)$ is negative, i.e.

$$(v_{j+1} - v_j) \times (p_i - v_j) = (x_i - \zeta_j)(\eta_{j+1} - \eta_j) - (\zeta_{j+1} - \zeta_j)(y_i - \eta_j) \leq 0 \quad (19)$$

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 p_i i = 1, ..., N_c, and for all couples of

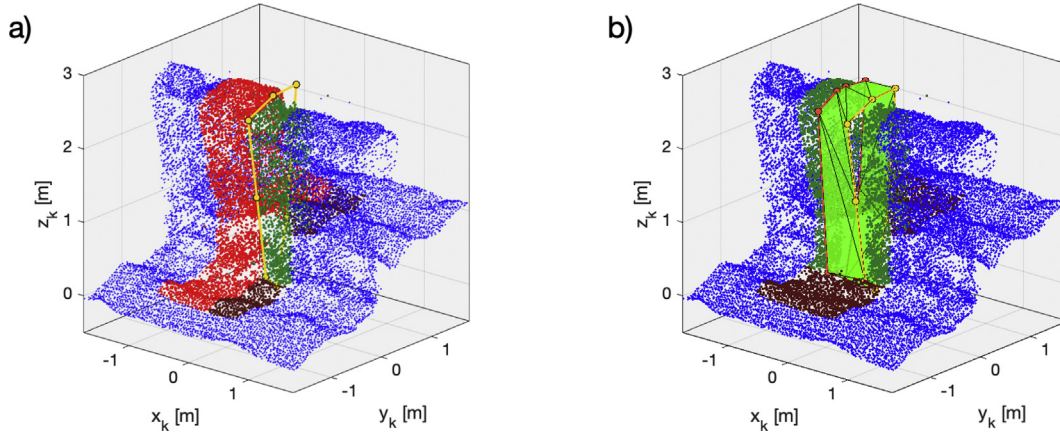**Fig. 7 − Low complexity triangulated mesh generation: (a) c-gons $\mathcal{P}[V_{268}^{\{LOC_{268}\}}]$ (cyan) and $\mathcal{P}[V_{267}^{\{LOC_{267}\}}]$ vertices (red) and (b) the generated low complexity mesh $\mathcal{M}_{268}$. Animation of this Figure can be find in the additional material in the electronic version of this manuscript.**

vertices $v_j, v_{j+1}, j = 1, \ldots, c-1$. The last equation takes into account the line passing through the two vertices v_c, v_1. Since Eq. (16) was found to be bilinear and, hence, non-convex, 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. 6c), always in position $v_1 = [00]$, $v_2 = [w^+ h^+]$ and $v_c = [w^- h^-]$, allowing to remove the last constraint in Eq. (15), which would be automatically satisfied.

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

$$V_k^{\{LOC_k\}} = \{v_i = [0, \zeta_i, \eta_i]^T \mid v_i = [\zeta_i, \eta_i]^T \in V^{2D}\} \quad (20)$$

and then in the absolute $\{WGS84\}$ in order to make them suitable for the final processing step to determine a low complexity triangulated mesh generation. In Fig. 6c, a c-gon $\mathcal{P}^7$ enclosing the given set of points $\mathcal{C}_{268}^{2D}$ with vertices V_{268}^{2D} is represented, with fixed vertices $v_1 = [0\ 0]$, $v_2 = [w_{268}^+ h_{268}^+] = [0.22 0.87]$ and $v_7 = [w_{268}^- h_{268}^-] = [-0.28 0.26]$, whereas its representation in the 3D reference system $\{LOC_k\}$ can be observed in Fig. 6d.

2.3.3. Triangulated mesh building

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

$$F_{k-1,k+1} = \{(v_{k-1,i}, v_{k-1,i+1}, v_{k,i}), (v_{k-1,i+1}, v_{k,i+1}, v_{k,i}) \ \forall\ i = 1, \ldots, c\} \quad (21)$$

where V_{k-1} and V_k are the sets of mesh vertices, described in the previous section, and $F_{k-1,k}$ is the set of triangular faces of the mesh between them (Fig. 7b). A triangular face is defined as triplets of points v, so that $F_{k-1,k}$ can be expressed as

$$F_{k-1,k+1} = \{(v_{k-1,i}, v_{k-1,i+1}, v_{k,i}), (v_{k-1,i+1}, v_{k,i+1}, v_{k,i}) \ \forall\ i = 1, \ldots, c\} \quad (22)$$

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. 7b. A sample portion of raw 3D point cloud $\mathcal{PC}^{\{WGS84\}}$ (blue dots) and low complexity triangulated 3D mesh model $\mathcal{M}$, generated by linking polygon vertices $\mathcal{P}[V_k]$ between adjacent sections $\mathcal{S}_k$ can be observed in Figs 2a and c, respectively. The procedure described in the previous sections, was repeated along the vine row model for all vine row section $\mathcal{S}_k$ with $k \in \mathcal{K}$.

Table 1 − Indices for quality score $\mathcal{Q}$ computation of mesh model $\mathcal{M}$.

Name	Description	Definition
good-modelling index G_k	volume of $\mathcal{C}_k$ properly modelled in $\mathcal{M}_k$	$G_k = V_{ref}^{-1} \cdot \int_{t_k}^{t_{k+1}} A_g(\gamma(t))\gamma'(t)dt$
under-modelling index U_k	volume of $\mathcal{C}_k$ not modelled in $\mathcal{M}_k$	$U_k = V_{ref}^{-1} \cdot \int_{t_k}^{t_{k+1}} A_u(\gamma(t))\gamma'(t)dt$
over-modelling index O_k	volume of $\mathcal{M}_k$ not present in $\mathcal{C}_k$	$O_k = V_{ref}^{-1} \cdot \int_{t_k}^{t_{k+1}} A_o(\gamma(t))\gamma'(t)dt$
complexity reduction index R_k	storage space reduction of $\mathcal{M}_k$ compared to $\mathcal{C}_k$	$R_k = 10^{-1} \cdot \left(1 - \dfrac{card(V_k) - 3 \cdot card(F_{k-1,k})}{card(\mathcal{C}_k) - card(\mathcal{C}_k \cap \mathcal{C}_{k-1})}\right)$
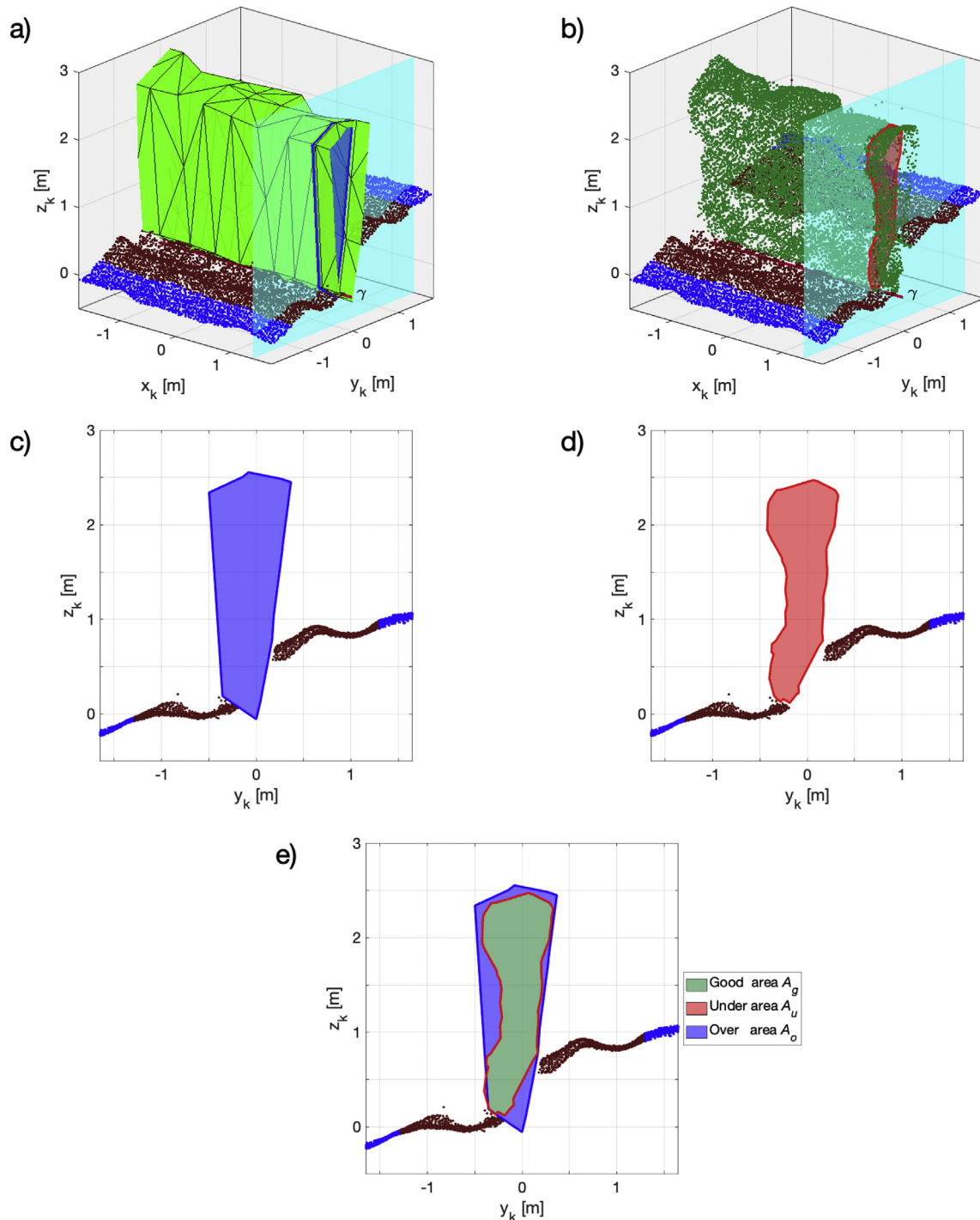
Fig. 8 — Model quality indices evaluation: (a—c) model $\mathcal{M}_k$ section; (b—d) points $\mathcal{C}_k$ envelope on line γ perpendicular plane; and (e) areas A_g, A_u and A_o, determined by comparing the two polygon in (c—d). Animation of this Figure can be find in the additional material in the electronic version of this manuscript.

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 $\mathcal{S}_k$ and $\mathcal{S}_{k+1}$ 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
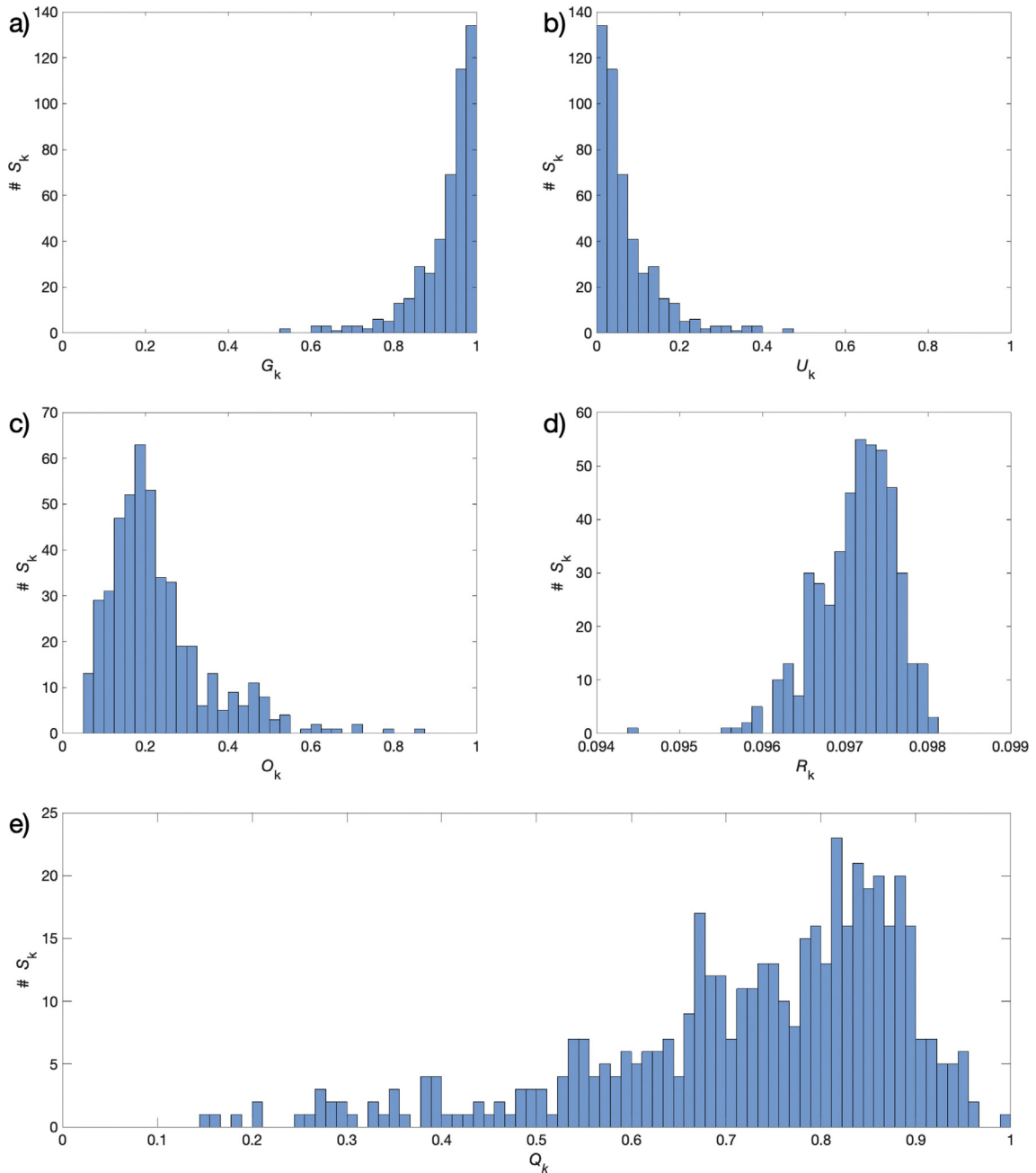
Fig. 9 – Model quality indices results histogram obtained by the model $\mathcal{M}^{[0.4,\ 0.25,\ 7]}$ (with $a = 0.4$ m, $b = 0.25$ m and $c = 7$): (a) *good modelling* G_k; (b) *under modelling* U_k; (c) *over modelling* O_k; (d) *complexity reduction* R_k indices; and (e) *quality score* $\mathcal{Q}_k$.

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 $\mathcal{Q}$ 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 $\mathcal{Q}$ 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

$$\mathcal{Q}_k = G_k - (U_k + O_k) + R_k \qquad (23)$$

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 1, where A_g, A_u and A_o are the areas derived from the intersection of $\mathcal{M}_k$ with a plane perpendicular to line γ (Fig. 8) and where V_{ref} is the $\mathcal{C}_k$ envelope volume.

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 1280 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 $\overline{\mathcal{Q}} = 0.71$, and a standard deviation of $\sigma_Q = 0.19$ (Fig. 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 $\mathcal{S}_k$, are reported in Fig. 9. The *good modelling* index G_k had an overall mean value of $\overline{G}_m = 0.92$ 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 $\overline{U}_m = 0.07$ and $\overline{O}_m = 0.23$, 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 $\overline{R}_m = 0.09$ and a very small standard deviation of $\sigma_R = 0.05 \cdot 10^{-2}$.

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 $\mathcal{PC}^{\{WGS84\}}$ 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. 2c.

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

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.

Acknowledgments

REFERENCES

Agisoft©. from http://www.agisoft.com. (Accessed November 2019).

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. https://doi.org/10.1016/j.biosystemseng.2018.11.005.

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering, 149*, 94−111. https://doi.org/10.1016/j.biosystemseng.2016.06.014.

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.

Boland, R. P., & Urrutia, J. (2000). *Polygon area problems.* 12th Canadian Conference on computational geometry.

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. https://doi.org/10.1007/s11119-019-09643-z.

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. In *2019 IEEE international workshop on metrology for agriculture and forestry (MetroAgriFor)* (pp. 62–67). https://doi.org/10.1109/MetroAgriFor.2019.8909219.

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. https://doi.org/10.1016/j.compag.2018.10.005.

Comba, L., Biglia, A., Ricauda Aimonino, D., Tortia, C., Mania, E., Guidoni, S., et al. (2020). Leaf Area Index evaluation in vineyards using 3D point clouds from UAV imagery. *Precision Agriculture, 21*, 881–896. https://doi.org/10.1007/s11119-019-09699-x.

Comba, L., Ricauda Aimonino, D., & Gay, P. (2016). Robot ensembles for grafting herbaceous crops. *Biosystems Engineering, 146*, 227–239. https://doi.org/10.1016/j.biosystemseng.2016.02.012.

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. https://doi.org/10.1016/j.compag.2020.105394.

de Berg, M., van Kreveld, M., Overmars, M., & Cheong, O. (2000). *Computational geometry: Algorithms and applications.* Springer.

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. https://doi.org/10.1016/j.biosystemseng.2020.02.014.

Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science, 327*, 828–831. https://doi.org/10.1126/science.1183899.

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. https://doi.org/10.3390/su9050728.

Grimstad, L., & From, P. J. (2017). Thorvald II - a modular and Re-configurable agricultural robot. *IFAC-PapersOnLine, 50*, 4588–4593. https://doi.org/10.1016/j.ifacol.2017.08.1005.

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. https://doi.org/10.1016/j.robot.2015.11.009.

Han, J. (2018). An efficient approach to 3D path planning. *Information Sciences, 478*, 318–330. https://doi.org/10.1016/j.ins.2018.11.045.

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*, 67–74. https://doi.org/10.1016/j.eja.2019.04.001.

Kassler, M. (2001). Agricultural automation in the new millennium. *Computers and Electronics in Agriculture, 30*, 237–240. https://doi.org/10.1016/S0168-1699(00)00167-8.

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. https://doi.org/10.3390/rs11040436.

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. https://doi.org/10.1016/j.biosystemseng.2016.08.026.

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. *Computers and Electronics in Agriculture, 135*, 300–311. https://doi.org/10.1016/j.compag.2017.02.017.

Maes, W. H., & Steppe, K. (2019). Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends in Plant Science, 24*, 152–164. https://doi.org/10.1016/j.tplants.2018.11.007.

Mathworks Matlab®. from https://uk.mathworks.com/help/signal/examples/peak-analysis.html, (2020)–. (Accessed November 2019).

Mathworks Matlab®. from https://www.mathworks.com/products/matlab.html, (2020)–. (Accessed November 2019).

Mathworks Matlab®. from https://uk.mathworks.com/discovery/genetic-algorithm.html, (2020)–. (Accessed November 2019).

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. https://doi.org/10.3390/s20092530.

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. https://doi.org/10.3390/s150203671.

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. https://doi.org/10.1016/j.compag.2018.09.010.

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. https://doi.org/10.1016/j.biosystemseng.2015.12.004.

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

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. https://doi.org/10.1007/s11119-017-9532-7.

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. https://doi.org/10.1016/j.jclepro.2019.04.347.

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. https://doi.org/10.1080/22797254.2017.1308234.

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. In *European conference (10th) on precision agriculture* (pp. 465–470). https://doi.org/10.3920/978-90-8686-814-8_57.

Reina, G., Milella, A., & Galati, R. (2017). Terrain assessment for precision agriculture using vehicle dynamic modelling. *Biosystems Engineering, 162*, 124–139. https://doi.org/10.1016/j.biosystemseng.2017.06.025.

Reza, M.d. 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. https://doi.org/10.1016/j.biosystemseng.2018.09.014.

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, 22*, 2420–2427. https://doi.org/10.1109/TMECH.2017.2663436.

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. https://doi.org/10.1016/j.compag.2015.12.014.

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. https://doi.org/10.1002/jsfa.9545.

Sozzi, M., Kayad, A., Giora, D., Sartori, L., & Marinello, F. (2019). Cost-effectiveness and performance of optical satellites constellation for Precision Agriculture. In *European conference (10th) on precision agriculture* (pp. 501–507). https://doi.org/10.3920/978-90-8686-888-9_62.

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. https://doi.org/10.1016/j.compag.2019.05.002.

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. https://doi.org/10.1016/j.compag.2018.08.043.

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. https://doi.org/10.1111/wre.12255.

Van Henten, E. J., Bac, C. W., Hemming, J., & Edan, Y. (2013). Robotics in protected cultivation. *IFAC Proceedings, 46*, 170–177. https://doi.org/10.3182/20130828-2-SF-3019.00070.

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. https://doi.org/10.1016/j.biosystemseng.2014.10.003.

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*, 82–90. https://doi.org/10.1016/j.compag.2019.03.017.

Weiss, U., & Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and Autonomous Systems, 59*, 265–273. https://doi.org/10.1016/j.robot.2011.02.011.

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. https://doi.org/10.1016/j.jag.2018.10.006.

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming – a review. *Agricultural Systems, 153*, 69–80. https://doi.org/10.1016/j.agsy.2017.01.023.

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. https://doi.org/10.1016/j.compag.2019.03.037.

Zeybek, M., & Şanlıoğlu, İ. (2019). Point cloud filtering on UAV based point cloud. *Measurement, 133*, 99–111. https://doi.org/10.1016/j.measurement.2018.10.013.