



Deliverable report

SHEET

PROJECT ID	DATE SUBMITTED
40661	MM/DD/YYYY
Working package (WP) objective - Number (No)	
WP2: The detection of fruit is a prerequisite for gaining fruit temperature data in the tree canopy by means of remote sensing. The fruit detection will be approached by means of thermal imaging directly, but a rather high measuring uncertainty is assumed. Consequently, more advanced methods, namely LiDAR laser scanning and RGB-D imaging will be employed to detect the fruit. In parallel, the thermal imaging will be carried out. The objective of the present deliverable is the alignment of geometric fruit data and thermal data on these fruit to provide the fruit temperature data with high spatial resolution. Reference measurements will capture readings by microclimate sensors and manual IR sensors. Additionally, the potential of the RGB-D imaging will be tested for detecting the fruit damage degree directly at the tree.	

Project Deliverables (D)

Deliverable No	Description
D2.1	Alignment code considering fruit 3D points and temperature data is available for apple.
D2.2	Codes for fruit detection using RGB-D and laser scanner are available for the three fruits.

D2.3	Fruit position and temperature data (preliminary) are available as input for further modelling in apple.
------	--

Due date

Deliverable No	Due date in month (M)
D2.1	M3-31/05/2021
D2.2	M6-28/02/2022
D2.3	M12-28/02/2022

Partners

Deliverable No	Lead partner (Contributors)
D2.1	ATB
D2.2	ATB, (UNIBO)
D2.3	ATB

Deliverable No D2.1

Deliverable hypothesis

Deliverable No	Hypothesis
D2.1	<ul style="list-style-type: none">• The thermal cameras camera matrix and distortion, as well as position and orientation relative to the LiDAR sensor can be defined through intrinsic and extrinsic calibration using a calibration target.• Alignment of 3D point and temperature value can be performed using intrinsic and extrinsic calibration parameters

Deliverable Description

Deliverable No	Description
D2.1	Alignment code considering fruit 3D points and temperature data is available for apple.

Lidar to Thermal Camera Calibration

Sensor setup

An aluminium frame carrying the sensors was mounted on a rigid linear tooth-belt conveyor system (Module 115/42, IEF Werner, Germany) of 800 mm length, employing a servo positioning controller (LV-servoTEC S2, IEF Werner, Germany). A outdoor 2D LiDAR sensor (LMS-511, Sick AG, Waldkirch, Germany) was mounted vertically on the metal frame at 0.8 m above the ground level. A thermal camera (A655sc 80°, FLIR Systems Inc., MA, USA) was placed, at about 0.2 m distance, above the laser scanner (Fig.1).

The LiDAR and the thermal camera were connected via Ethernet to a laptop with software developed in LabVIEW (version NXG 5.1, National Instruments, Texas, USA) for data acquisition. The LiDAR sensor was configured with a 0.1667° angular resolution, 25 Hz scanning frequency and a scanning angle of 180°. The thermal camera has an image resolution of 640 × 480 pixels at 50 Hz and a thermal resolution <0.05 °C. In parallel, the positioning controller of a linear conveyor was connected to the same computer with a RS-232 serial port using the S2 Commander software (version 4.1.4201.1.1, IEF Werner, Germany) for configuration and operation. The linear conveyor was configured at 20 mm s⁻¹ (±0.05 mm accuracy) forward speed.

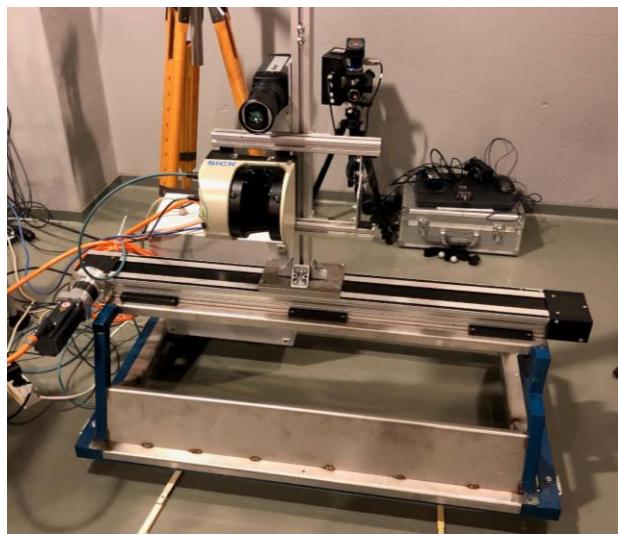


Fig. 1. Representation of the sensor-frame system showing LiDAR laser scanner and thermal camera

An active calibration pattern with clearly defined heat sources ($n = 30$) was constructed and used for calibrating the thermal camera (Fig. 2). The 12 Volt lamps, each with a glass-lightbulb diameter of 4 mm, served as heat source. The overall size of the board is 500 mm (width) \times 600 mm (height). The board target was scanned by means of the sensor frame using the linear conveyor.

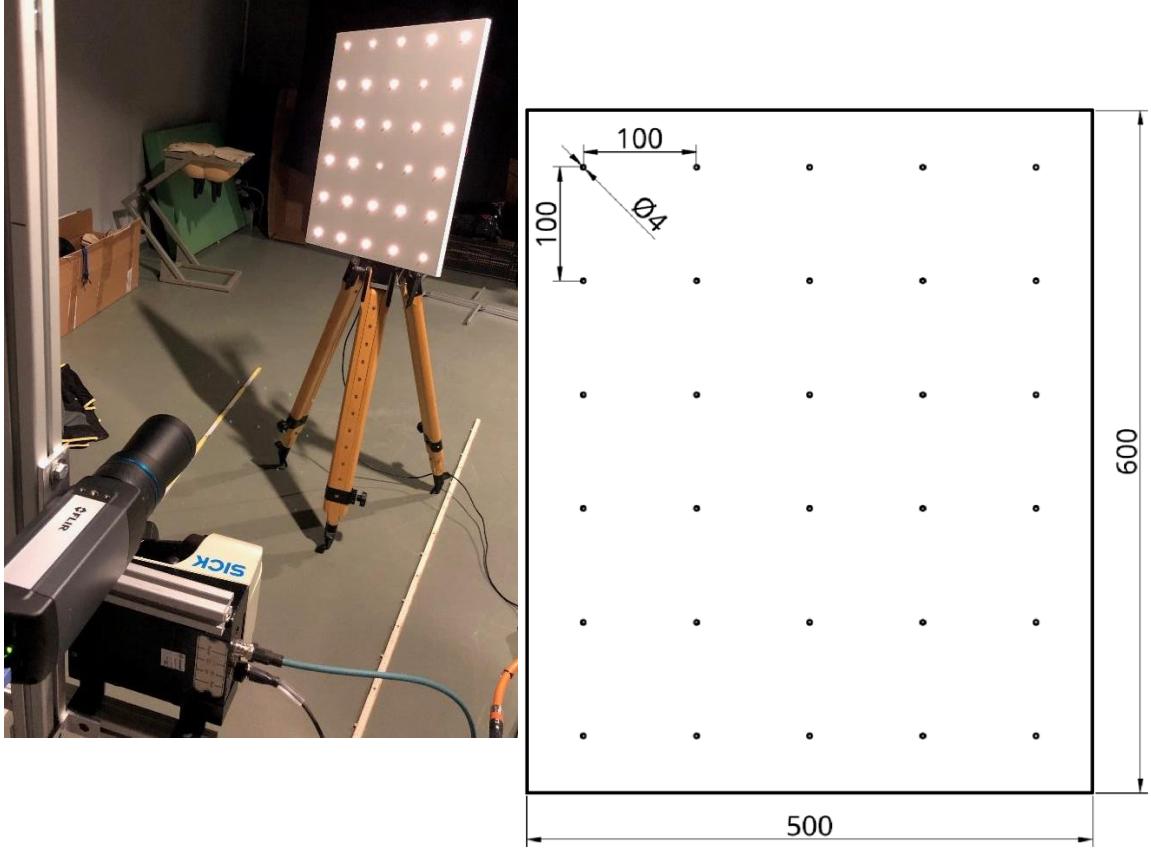


Fig. 2. Active board target of known dimensions (500 x 600 mm) with lightbulbs ($n = 30$).

Data processing

Intrinsic calibration of the thermal camera

Each sensor perceives the world in its device-specific local coordinate system. To join the perceived information we need the specific parameters of these coordinate systems. The thermal camera has unique parameters that define how a point (X, Y, Z) in world coordinates is projected onto the image plane. Given the focal length (fx, fy) of the camera, the camera center (cx, cy) image coordinates (x, y) are calculated as:

$$[xy] = [fx 0 cx 0 fy cy 0 0 1] \begin{bmatrix} \frac{x}{z} & \frac{y}{z} & 1 \end{bmatrix} \quad (1)$$

Given the radial distortion coefficients $k1, k2, k3$ and the tangential distortion coefficients $p1, p2$ and $r = \sqrt{x^2 + y^2}$, the corrected image points (xc, yc) are calculated as:

$$(xcyc) = (x(1 + k1r^2 + k2r^4 + k3r^6) + 2p1y + p2(r^2 + 2x^2)y(1 + k1r^2 + k2r^4 + k3r^6) + p1(r^2 + 2y^2) + 2xp2) \quad (2)$$

Identifying the heat sources in the image enables us to perform intrinsic calibration in the same way as for optical cameras (Fig. 3). To this end, the raw thermal image (Fig. 3a) is first scaled to its temperature range

(Fig. 3b) to make the temperature blobs more visible. Next, OpenCVs (<https://opencv.org>) SimpleBlobDetector is used to find the necessary keypoints in the image with sub-pixel accuracy. If the number of found keypoints is equal to the number of lightbulbs, the keypoints are sorted according to the pattern shown in Fig. 3c, starting in the top right corner. The aforementioned camera parameters are then calculated using OpenCVs calibrateCamera function.

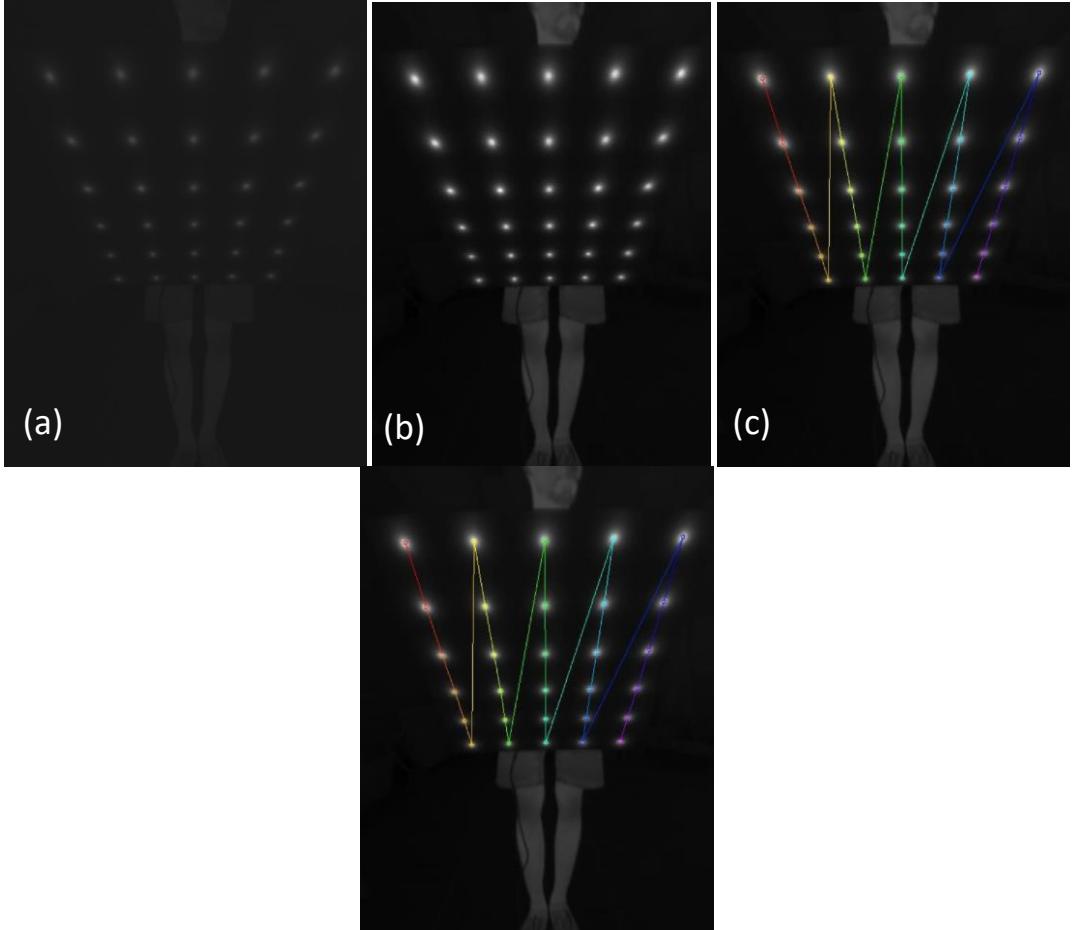


Fig. 3. (a) Raw thermal image, (b) scaled to temperature range and (c) detected and sorted lightbulbs

Extrinsic calibration – Thermal camera and LiDAR

After calculating the internal camera parameters, the rotation and translation between the local camera coordinate system and the coordinate system of the LiDAR sensor need to be determined. The three rotational and three translational parameters are known as the extrinsic. Once all the points are in the camera coordinate system, the projection to the image can be defined up to a factor s using equation (3):

$$s [xy1] = [fx\ 0\ cx\ 0\ fy\ cy\ 0\ 0\ 1] [r_{11} r_{12} r_{13} t_1 r_{21} r_{22} r_{23} t_2 r_{31} r_{32} r_{33} t_3] [XYZ] \quad (3)$$

Suppose there are n images of the calibration pattern and m planar points on the pattern considering the distortions as independent and identically distributed noise than the maximum likelihood estimate of the transformation between the scanner and camera coordinate system is obtained by minimizing

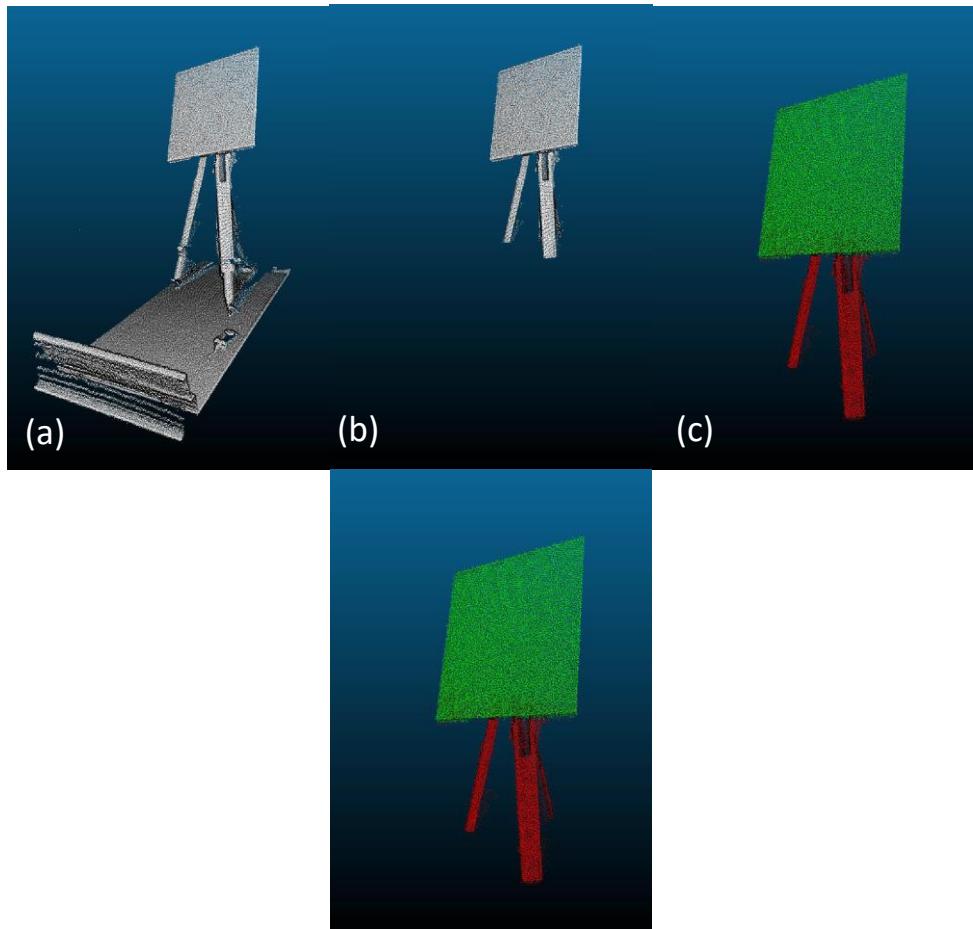
$$\sum_{i=1}^n \sum_{j=1}^m \|p_{i,j} - p^{(A, D, R_i, t_i, P_j)}\|^2 \quad (4)$$

where A is the intrinsic matrix, R_i the rotation matrix, t_i the translation vector, and D the distortion parameters. $p_i(A, D, R_i, t_i, P_j)$ defines the projection of point P_j in image i , according to equation (1) and (2). This approach assumes that we have a number of points that are identifiable in both the laser scan and the image. For this purpose we attach the calibration pattern onto a board. For the thermal camera light bulbs arranged in a regular grid pattern. The position of the points of these patterns are known. Algorithm 1 detects the points in a laser scan.

Algorithm 1. Calibration pattern detection in a laser scan

Require: point cloud, specification of calibration pattern

- 1: discard points outside the area of the expected board (Fig. 4b)
- 2: find the most prominent plane using random consensus (RANSAC) algorithm (Fig. 4c)
- 3: project a generated plane model into the center of the detected plane
- 4: use ICP to fit the plane model to the data points (Fig. 4d)
- 5: **return** position of the lightbulbs according to iterative closest point (ICP) result (Fig. 4e)



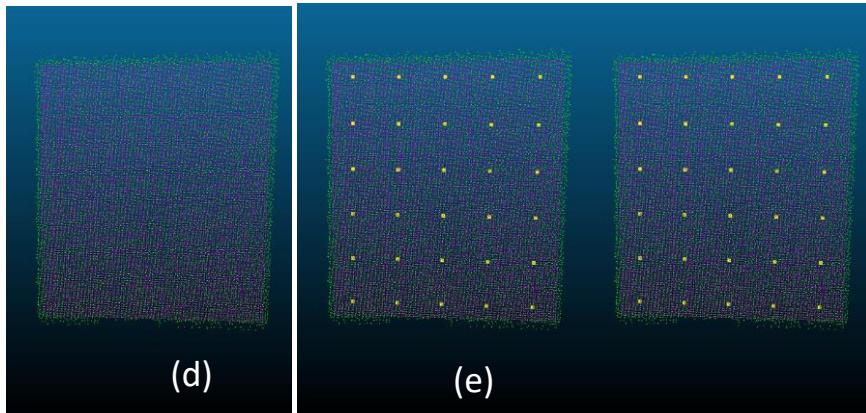


Fig. 4. (a) Raw point cloud, (b) Cropped point cloud, (c) Prominent plane using RANSAC, (d) finely registered model of calibration to point cloud using ICP and (e) return position of the lightbulbs

3D to 2D Projection and Color Mapping

During the data acquisition phase, laser scans and images are acquired simultaneously. After determining the relations between scanner and thermal camera in the calibration step this relation is used directly to color the point cloud.

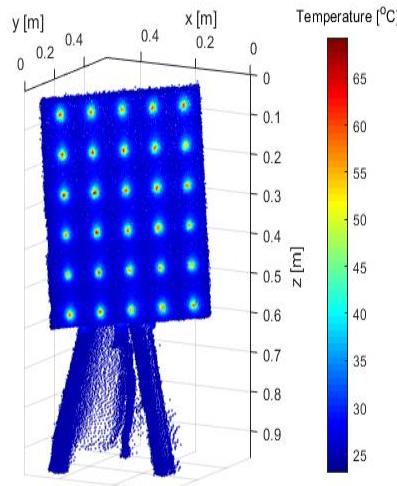


Fig. 5. 3D thermal point cloud

Python Code HOWTO

General

- The whole code is based on python
- The following libraries are needed
 - numpy
 - opencv
 - glob
 - argparse
 - pandas

- open3d
- Currently (26.07.21) it is **mandatory** to use python 3.8, since open3d **does not** support python 3.9, therefore I highly suggest creating a python virtual environment in order not to mess with everything else

Programs

1. intrinsics.py

Info

This program uses recorded images of a lightbulb checkerboard pattern to calculate the intrinsics of the thermal camera, it can also use checkerboard images to calibrate any regular camera.

Usage

The currently available parameters are

```
usage: intrinsics.py [-h] [-c arg] [-x arg] [-y arg] [-p arg]
                     [-d arg] [-t arg]

optional arguments:
  -h, --help            show this help message and exit
  -c arg, --camera arg  Type of camera. 'thermal' or 'color'.
  -x arg, --boardX arg  Number of corners in x direction (chessboard).
  -y arg, --boardY arg  Number of corners in y direction (chessboard).
  -p arg, --path arg    Path to calibration images.
  -d arg, --debug arg   Enable debug mode, which prints more info to the
                       command line and previews images.
```

Hints

There are a few key rules for taking the images for the calibration

- set the focus for the desired distance and **DON'T** change it afterwards --> disable autofocus
- for regular cameras, set the aperture and don't change it afterwards
- record images of the pattern at different orientations and positions within the image frame
- record between 50 and 150 images (the more the better, since some outliers will be filtered out)
- when finished, the calibration matrix as well as the distortion vectors are printed to the command line
- also the RMS error is printed to the command line and should not exceed 1.5

Output

A file called **intrinsics.npz** is saved in the folder containing the calibration images, in other words what was given via the command line argument -p or –path

2. extrinsics.py

Info

This tool reads LiDAR 3D point cloud and thermal image data and tries to estimate the extrinsics of the setup.

Data Structure

When recording the data, make sure that the structure is as follows:

```
data
  +- scan_01
  | +- file01.txt
  | +- file02.txt
  | +- ...
  | +- DATE_lidar.txt
  |
  +- scan_02
  | +- file01.txt
  | +- file02.txt
  | +- ...
  | +- DATE_lidar.txt
  +- ...
```

- The folder names **must** contain **scan**
- The LiDAR filename must contain **lidar** and the DATE as %y%m%d, e.g. 210723_lidar.txt
- The thermal image .txt file names must contain the date and time as %Y%m%d%H%M%S%f, e.g. 2021072311542825134.txt

Usage

The currently available parameters are:

```
usage: extrinsics.py [-h] [-p PATH] [-i INTRINSICS] [-n PATTERN] [-d DEBUG]
```

optional arguments:

- h, --help show this help message and exit
- p PATH, --path PATH Path to data folder.
- i FILE, --intrinsics FILE
Path to the intrinsics calibration file.
- d arg, --debug arg Enable debug mode, which prints more info to the command line and previews data.

Hints

There are a few key rules for taking the data for the calibration

- focus from intrinsic calibration must still be unchanged!
- record images of the pattern at different orientations and positions within the scan area and be sure to not cut off sections of the board
- around 20 to 30 different pattern positions should be recorded for calibration
- when finished, the extrinsic calibration matrix is printed to the command line

Output

A file called **calibration.npz** is saved in the folder containing the data folders, in other words what was given via the command line argument -p or –path as PATH. Also, the calibration data is fused and gets saved into PATH.

3. **colorize.py** still work in progress

The data is colored based on existing calibration files.

The way it works in short:

- Each point of the cloud is projected onto the image plane
- For each resulting pixel coordinate, the corresponding temperature is taken from the pixel and assigned to the point

Hypotheses were achieved

Thermal's camera position was defined utilizing an active calibration target with lightbulbs. The alignment of LiDAR's 3D point cloud with thermal image was performed with the extrinsic calibration between the sensors. Therefore, the 3D point cloud was colourized with temperature information.

The assessment of calibration and alignment methodology will perform also in field conditions in order.

Availability/access of the code

A first version of HowTo.md will be available online in Github.com by the end of October 2021. The version will include the tools for intrinsic, extrinsic calibration of sensors, and the alignment between 3D point cloud and the thermal images. Note on the availability/access of the code.

Usage by other groups

In case of possible usage by other working groups, the code should be reference

The Note on possible usage by other work groups.

Deliverable No D2.2

Deliverable hypothesis

Deliverable No	Hypothesis
D2.2	<ul style="list-style-type: none">• Apple fruit can be segmented from LiDAR and RGB-D 3D point cloud of trees in the orchard.• Cherry fruit can be segmented from 3D point cloud of trees in the orchard.<ul style="list-style-type: none">• Grape can be segmented from 3D point cloud of trees in the orchard.

Deliverable Description

Deliverable No	Description
D2.2	<ul style="list-style-type: none"> Codes for fruit detection using RGB-D and laser scanner are available for the three fruits.

Apple segmentation by means of LiDAR laser scanner

Data Acquisition and Pre-Processing

A metal frame mounted on a tractor was used to carry the sensors along the tree rows (Tsoulias et al., 2019). A mobile outdoor 2D LiDAR sensor (LMS-511, Sick AG, Waldkirch, Germany) was mounted vertically on the metal frame at 1.6 m above the ground level. The LiDAR sensor was configured with a 0.1667° angular resolution, 25 Hz scanning frequency and a scanning angle of 190°. The sensor frame was driven along the rows on both sides of the trees with an average speed of 0.13 m s⁻¹. The measured objects, hit by the laser beam, were assumed as perfectly diffuse reflectors (Lambertian), excluding the influence of incidence angle. Thus, the apparent signal was considered as an approximation of hemispherical reflectance. Board targets, coated with barium sulphate (BaSO₄, CAS Number: 7727-43-7, Merck, Germany) for white (R_{white}) and urethane (S black, Avian Technologies, New London, NH, USA) for black (R_{black}) reference were applied to calibrate the backscattered intensity (R_{ToF}) of the LiDAR, obtaining the R_{ToF} [%] at 905 nm for each point in the 3D point cloud (eq.5).

$$R_{ToF} = \frac{(R | measured - R_{dark})}{(R_{white} - R_{dark})} \quad (5)$$

An inertial measurement unit (IMU) (MTi-G-710, XSENS, Enschede, the Netherlands) was placed 0.24 m aside from the LiDAR to monitor the 3D tilt of the tractor. The accuracy of the sensor was 1.0° root mean square error (RMSE) for the heading at static and dynamic mode, and 0.25° RMSE and 1.0° RMSE for both pitch and roll in static and dynamic mode, respectively (Kooi, 2016). The 3D point cloud was georeferenced using a RTK-GNSS (AgGPS 542, Trimble, Sunnyvale, CA, USA) mounted above the laser scanner at the frame in 1.74 m height. The horizontal accuracy of the RTK-GNSS was ±25 mm +2 ppm and the vertical ±37 mm +2 ppm. A multi-thread software was developed in Visual Studio (version 16.1, Microsoft, Redmond, WA, USA) to acquire the data.

The 3D point cloud data was processed in the Computer Vision Toolbox™ of Matlab (2017b, Mathworks, Natick, MA, USA). A sparse outlier removal was applied to each point cloud pair to reject the points that were above the average distance using CloudCompare (2.10, GPL software, Paris, France). Moreover, the random consensus was performed to filter the points that belonged to the ground (Vázquez-Arellano et al., 2018). According to Tsoulias et al. (2019) the rigid translations and rotations were applied in each point of the point cloud, while the alignment of the pair tree sides was carried out with the iterative closest point algorithm, using a k-dimensional space to speed up the process.

Extraction of radiometric and geometric features

The information of the 3D local neighbourhood of points $P_i = [x_i, y_i, z_i]$ was determined using the k -nearest neighbours within a radius (r) equal to the D_{Manual} of apples (Hackel et al., 2016). The total number of the P_i set (N) was used to estimate the average $P = \frac{1}{N} \sum_{i=1}^N$ of the nearest neighbours. Thus, the covariance matrix (Cov) was built after mean centering by means of subtraction of P value from each P_i of the nearest neighbours set (eq. 6):

$$Cov(P) = \frac{1}{N} \sum_{i=1}^N (P_i - \bar{P}) \times (P_i - \bar{P})' \quad (6)$$

The Cov was decomposed based on the singular value decomposition, producing the eigenvalues (λ_1 , λ_2 , λ_3), which were sorted in ascending order $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and the corresponding eigenvectors. The eigenvalues, which represent the highest orthogonal variance of the matrix, provided the points dispersion within k -nearest neighbours describing the local spatial structure in 3D. The eigenvalues were normalized between 0 and 100, allowing the comparison of different clusters. Thus, the local geometry of each P_i in the point cloud was analyzed by means of eigenvalues to illustrate the spatial structure considering its linearity (L, eq. 7) and curvature (C, eq. 8):

$$L(P_i) = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (7)$$

$$C(P_i) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (8)$$

where L and C described the variation of linearity and curvatures for all points along the direction of the corresponding eigenvalues, respectively (Hackel et al., 2016; Lin et al., 2014). More specifically, the closer the values of L or C to 100, the higher the likelihood for the shape of points to be linear or curved, respectively. Therefore, L was used to segment the foliage and woody parts of the tree (branches, stem), while C performed to distinguish the apple points.

Steps of Apple Detection Methodology

- Define the range of R_{ToF} , C and L for the class of apple (R_A , C_A , L_A) and of woody parts (R_W , C_W , L_W) in defoliated trees (T_D), while the range of leaf class for the same features (R_L , C_L , L_L) was defined in the T_L .
- The thresholding was performed for each class by employing the exploratory analysis of normal distribution using the probability density function to define the threshold that will distinguish the 3D points of apples from leaves and woody parts.
- The value with the highest likelihood (mode) within the R_A , C_A , and L_A classes was used as a threshold (R_{th} , C_{th} , and L_{th}). Points which fulfilled the criteria of $L_A \leq L_{th}$, $C_{th} \leq C_A$, and $R_{th} \leq R_A$ were segmented and categorized as apples.
- A density-based scan algorithm (DBSCAN) was applied to find the point sets, using the mean diameter of apples that was found in each tree as a neighborhood search radius (ϵ) and the value 10 as a minimum number of neighbors. The value of 10 was applied resulting from manually run tests showing that less neighboring points result in random appearance of sets. The mean value of apparent reflectance intensity and of curvature in defoliated trees (R and C) were calculated to classify the clusters.
- The k-means algorithm defined the clusters of apple centers (ML). Furthermore, the maximum distance in x and y axes was considered as the diameter (D_L) of each point set. The main partitioning condition remained the same using the generated features from each cluster. Thus, the features of each T_L sub-cluster were iteratively extracted and re-evaluated until fulfilling the conditions. The number of partitions was used in the k-means algorithm to determine the centers of M_L sub-clusters. The only difference between analysing T_D and T_L was the use of the sphere in T_D to evaluate the eligibility of the cluster or the need to search for subclusters according to the estimated manually measured fruit diameter.

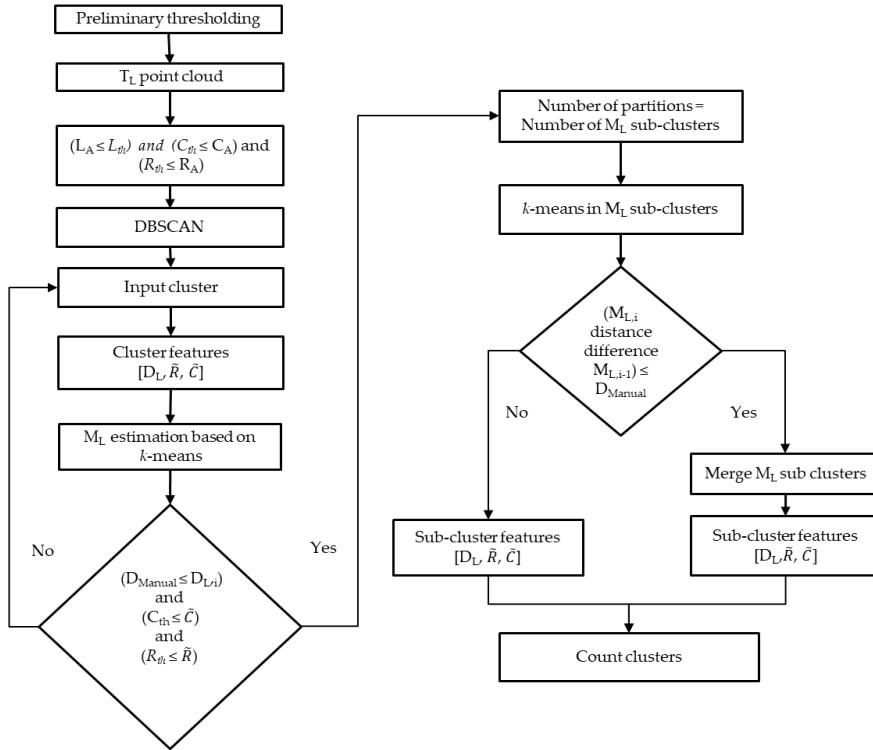


Fig. 6. Flowchart showing the protocol for apple detection and sizing using foliated (T_L) trees, starting with the threshold application in the registered point cloud pair, through the filtering and partitioning, until the counting of centres of clusters.

The algorithm resulted in maximum values of 88.2% precision, 91.0% recall, and 89.5 F1 score at harvest (DAFB₁₂₀).

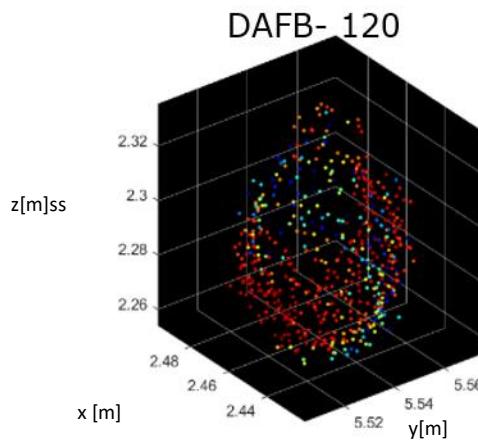


Fig. 7. Apple 3D point cloud at 120 days after full bloom.

More information can be found in: Tsoulias, N., Paraforos, D.S., Xanthopoulos, G. and Zude-Sasse, M., 2020. Apple shape detection based on geometric and radiometric features using a LiDAR laser scanner. *Remote Sensing*, 12(15), p.2481.

Apple and grape cluster detection by means of RGB-D camera

The fruit detection model investigated for RGB-D cameras exploit the application of a convolutional neural network (CNN) for image object detection. CNN are artificial intelligence algorithm those main goal is to solve high complex problems, generally related to image analysis/ computer vision solutions. The application of this CNN algorithm will be then used to detect fruits and extract 3D information from RGB-D collected data.

Data Acquisition, pre-processing

Data acquisition consisted in the creation of two datasets (one per each fruit species) on which to train a CNN algorithm to detect apple fruit and grape bunches (i.e., objects). The grape algorithm used a free online available “WGSD” dataset (<https://github.com/thasant/wgisd>; Santos *et al.*, 2020) made up of 300 pictures of different grape cultivars and conditions of shape, pose, illumination, phenological stage and focus. Regarding apple fruits, a brand-new dataset was created collecting 208 images of different cultivar (mainly Gala, Fuji and Pink lady®-Rosy glow) in different conditions of shape, pose, illumination, phenological stage. Different cameras sensors were utilized during the data collection (Smartphones: One Plus 2, Asus Zenfone 3, Huawei P20; Others: Intel Realsense D435i). For both the dataset, fruits were labelled in the image utilizing rectangular bounding boxes. While for grape-dataset this was already done, for apple-dataset labelling was done manually following a protocol defining occlusion (<80%), truncation (only visible part) and other rules with which to define if to label or not a fruit in an image. After labelling, apple-dataset, was including around 20000 apple fruits, while grape-dataset was including around 4500 grape clusters.

After datasets creation, each of these was divided in three sub-datasets called train-set, validation-set and test-set being respectively 70%, 20%, 10% of the whole image dataset. Both the datasets were then artificially augmented by copying original images and applying to each copy an image alteration. Alterations applied were flipping, mirroring, rotating, zooming plus cropping. Image augmentation increased dataset by approximately 20 times, obtaining around 4000 and 6000 images respectively for grape (>100k grape clusters) and apple (>100k apples).

Model training

The YOLO CNN algorithm “family” was chosen for fruit detection since it was recognized as well performing for fruit detection purposes, as reported by literature (e.g., YOLO: Bresilla *et al.*, 2019; YOLOv2-v3: Koirala *et al.*, 2019; YOLOv3: Tian *et al.*, 2019; YOLOv4: Parico and Ahamed, 2021). Particularly it was chosen the YOLOv5 CNN algorithm (<https://github.com/ultralytics/yolov5>), as it is one of the latest released version and can be considered a state-of-the-art algorithm. Considering the final idea to develop a smartphone app on which to use the outcome of the SHEET project the YOLOv5 model “m” (image size 640pixel) was chosen as it is a good compromise between model dimension and request of computational power to be used. The model was trained for 300 epochs on a dedicated workstation equipped with two graphic cards (i.e., NVIDIA RTX2080super), requiring 6-8 hours to complete the training per each model created (one for apple and one for grape). Models were trained on train-set and validation-set only, while test-set was not included in the training.

Model Performances(preliminary)

For apple (Fig. 8a), the trained model presented the best meanAveragePrecision (mAP) of 0.737 and the best F1-score of 0.74 for a confidence threshold of 0.37. The same model was then tested on the test-set, on images never utilized before, so to test the ability of the model in “generalize” its ability in detecting

fruits. Test-set results remained close to the training results showing best mAP of 0.734 and the best F1-score of 0.74 for a confidence threshold of 0.378 (Fig. 8a).

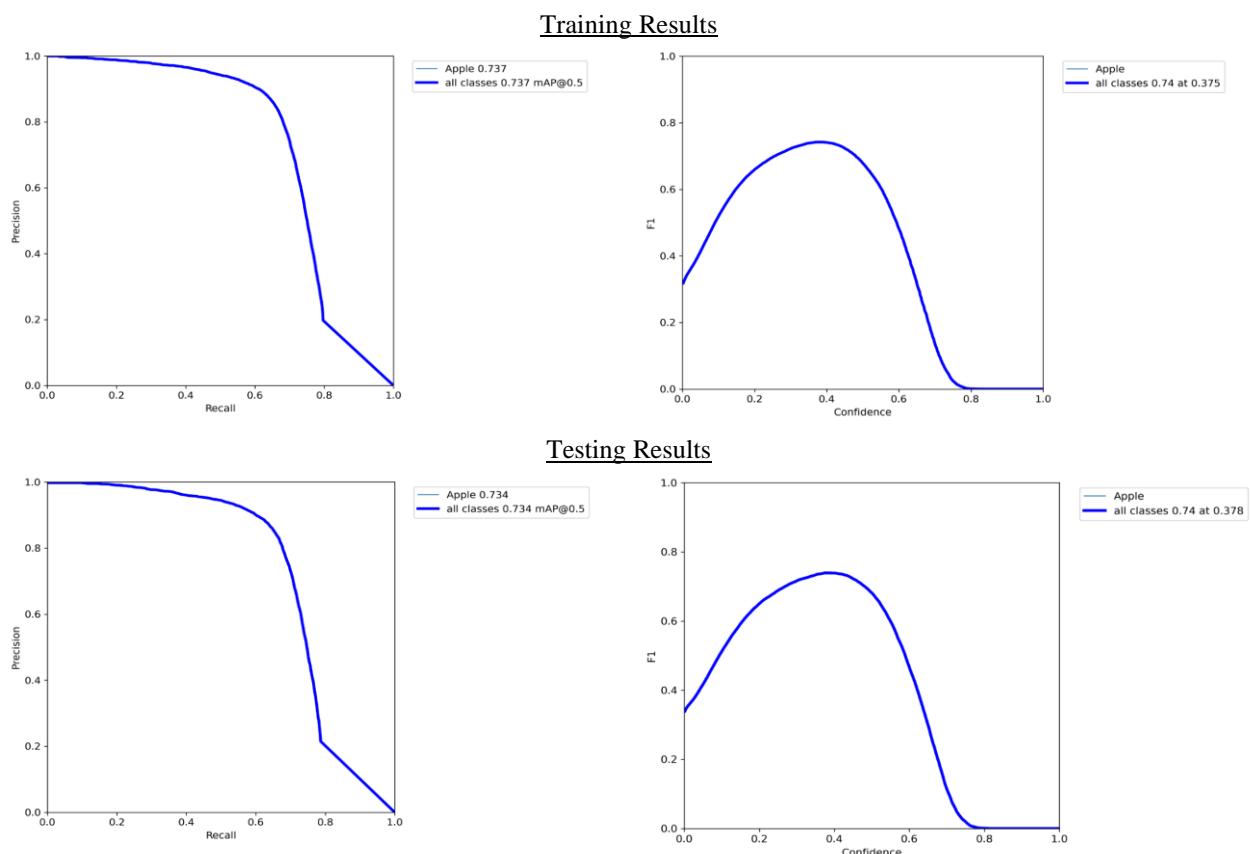


Fig. 8a. YOLOv5m model results for apple fruit detection.

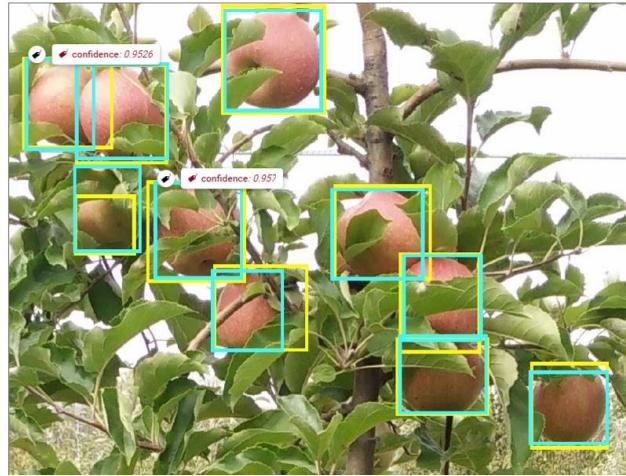
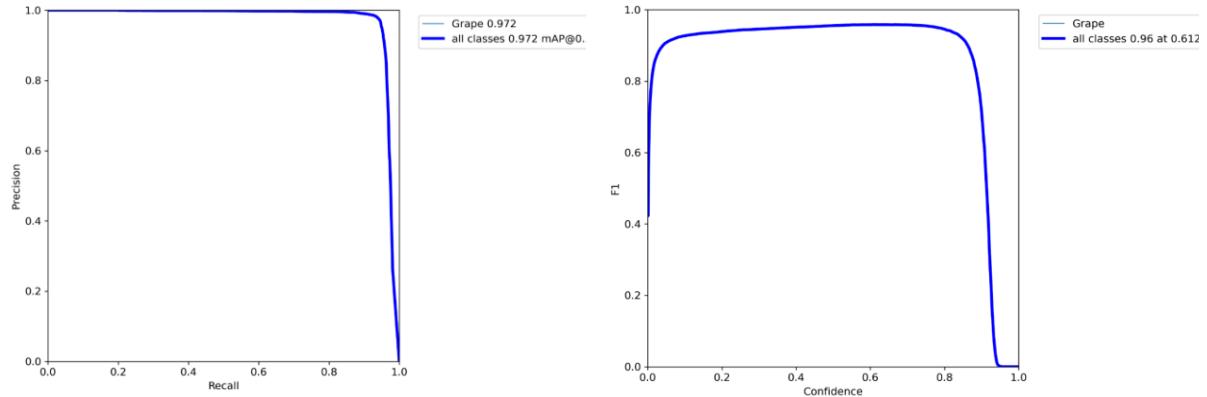


Fig. 8b. YOLOv5m model results for apple fruit detection. Yellow boxes are made manually, light blue boxes are results from the model inference.

For grape (Fig. 9a), the trained model presented the best mAP of 0.972 and the best F1-score of 0.96 for a confidence threshold of 0.612. The same model was then tested on the test-set, on images never utilized before, so to test the ability of the model in “generalize” its ability in detecting grape cluster. Test-set results remained close to the training results showing best mAP of 0.973 and the best F1-score of 0.96 for a confidence threshold of 0.57 (Fig. 9a).

Training Results



Testing Results

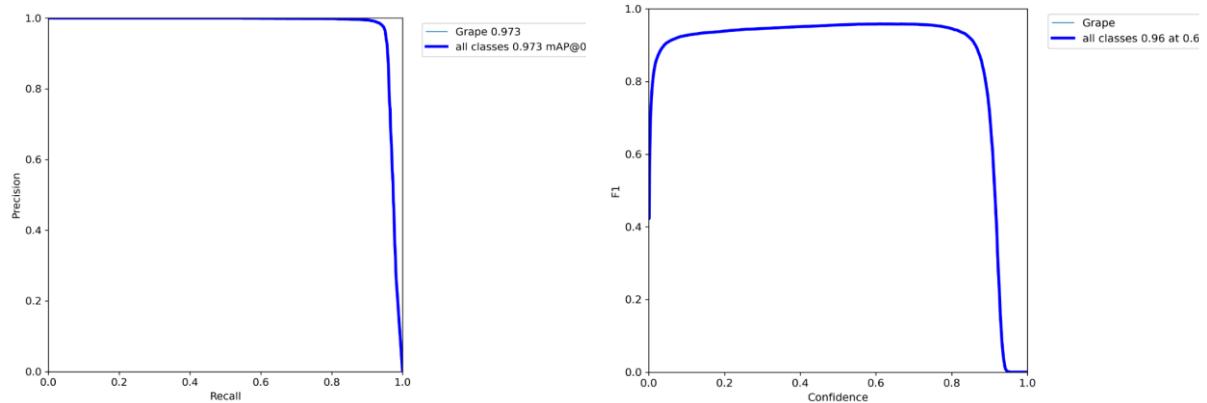


Fig. 9a. YOLOv5m model results for grape cluster detection.

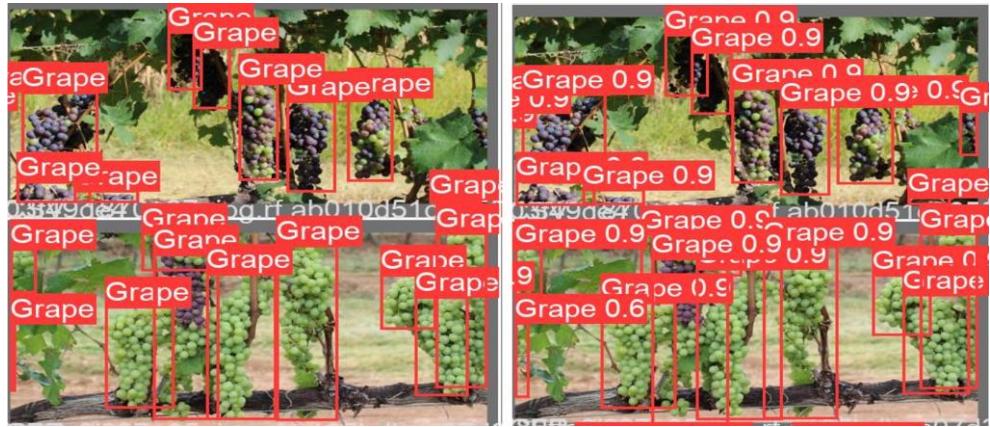


Fig. 9b. YOLOv5m model results for grape cluster detection. On the left, manually labelled expected result; on the right, results obtained from the model inference.

Grape cluster detection model seems to perform better than apple detection ones, and this seems mainly related to the different occlusion level present in the two different species. Apple shows many different levels of occlusion (i.e., 0 to 90%) that can alter the detection performances, while grape images tend to present a situation where clusters are generally, or highly visible (>70%) or highly occluded (>70%). This last situation improve model performance since highly visible cluster are easily well detected, while highly occluded clusters (>70%) are generally, correctly, undetected.

Point cloud detection of apple and grape clusters

The approach utilized to detect fruits on the RGB-D point cloud rely in the utilization of different functions included in the official Intel RealSense SDK (<https://github.com/IntelRealSense/librealsense>) of the RGB-D cameras utilized. The code is under development at the moment, but the following steps will be applied for extracting the 3D coordinates of each point present inside the 2D RGB fruit-detected bounding boxes:

1. Align RGB image and Depth map (i.e., matrix of depth information per each pixel) exploiting the function “align” that perform a pixel-to-pixel alignment of depth and RGB data; this allows to refer the same scene area, for both RGB and Depth map images, when utilizing RGB pixel coordinates.
2. Applying the CNN-based model for fruit detection on the “aligned” RGB image, extracting the list of all the detected fruits and their bounding boxes pixel coordinates.
3. Per each fruit detection in the list, a mean fruit distance will be extracted and all pixels inside the bounding boxes will be converted to points (x, y, z, based point cloud), exploiting the “deproject_pixel_to_point” function.
4. The point cloud created in this way, will be saved as it is the volume enclosing all the 3D points present inside the RGB fruit detection bounding boxes; improvement for the point cloud volume segmentation could be applied exploiting the mean fruit distance extracted so to reduce the 3D cuboid volume of the extracted point cloud

The result of this approach will be a series of single 2D bounding box-derived point cloud that should mainly represent 3D coordinates of the fruit detected. With this approach would be possible to reduce the point cloud volume to analyze with the above-mentioned LiDAR scanner segmentation approach. At the same time position information of the detected fruit can be extracted as well, with lower precision than LiDAR based approach, directly from aligned depth map analysis (exploiting a trigonometry approach) or from the 2D bounding box-derived point cloud.

Hypotheses were achieved

Apple fruit segmentation was performed using a LiDAR laser scanner. The results point to the high capacity of LiDAR variables [R_{ToF} , C, L] to localize fruit and estimate its size by means of remote sensing.

Fruit detection based on CNN algorithm were achieved with RGB-D cameras; point cloud extraction of the detected fruit area results possible with code under development.

Deliverable No D2.3

Deliverable hypothesis

Deliverable No	Hypothesis
D2.3	<ul style="list-style-type: none">• Evaluate the data fusion under laboratory and field conditions using a metal tree target and• Segment apple fruit surface temperature from 3D thermal tree point clouds.• Quantify fruit surface temperature variation at different position on the trees.

Deliverable Description

Deliverable No	Description
D2.3	Fruit position and temperature data (preliminary) are available as input for further modelling in apple.

The accurate projection of thermal information on the 3D point cloud requires calibration of the sensor frame system. More specifically, the process consists of two parts: (i) the intrinsic calibration of the thermal camera for determining camera matrix and distortion parameters, and (ii) the extrinsic calibration between camera and LiDAR coordinate system to define rotation and translation. The calibration tool chain is written in Python 3.8 (Python Software Foundation) and uses the OpenCV library (Bradski & Kaehler, 2008) for image processing and Open3D for point cloud processing. The methodology behind the calibration tool chain were described in D2.2.

When all intrinsic and extrinsic parameters determined, temperature values from the thermal image were assigned to the corresponding 3D points. All points in the point cloud were projected in the image plane. If the projected point lied within the image plane, the corresponding temperature value was assigned. Whereas, if the point was outside the plane, a value outside the temperature range of the dataset, in this case -10 °C, was assigned. This ensured that points outside of the field of view of the thermal camera were not omitted.

Segmenting apple temperature

After calibration, the phenotype system was mounted on the circular conveyor in order to scan the fruit trees from both sides (Fig.10). The temperature values assigned in the corresponding 3D point cloud were based on the extrinsic calibration. The LiDAR sensor configured with a 0.1667° angular resolution, 25 Hz scanning frequency, a scanning angle of 180 and a wavelength of 905 nm. Additionally, a thermal camera (A655sc, FLIR Systems Inc., MA, USA) placed at 0.2 m distance above the laser scanner. The camera has a spatial resolution of 640 × 480 pixels at 50 Hz, with a spectral range from 7.5 to 14 µm, an operational temperature range from -40°C to 150°C and a thermal resolution < 0.05°C. A lens (T198065, FLIR Systems Inc., MA, USA) with a focal length of 6.5 mm (diagonal 80°) is attached to the camera. The calibration was carried out in a room with no windows, controlled ventilation and temperature (15°C). The LiDAR and the thermal camera were connected via Ethernet to a laptop with software developed in LabVIEW (version NXG 5.1, National Instruments, Texas, USA) for data acquisition. The positioning controller of the linear conveyor connected to the same computer

with a RS-232 serial port using the S2 Commander software (version 4.1.4201.1.1, IEF Werner, Germany) for configuration and operation.

According to Tsoulias et al., (2019) rigid translations and rotations were applied on each point of the 3D point cloud, while alignment of pairing tree sides was carried out using ICP. The bivariate point density histogram was proposed to estimate the stem position of each tree ($n = 20$), while a cylindrical boundary was projected around the estimated stem positions to segment each individual tree.

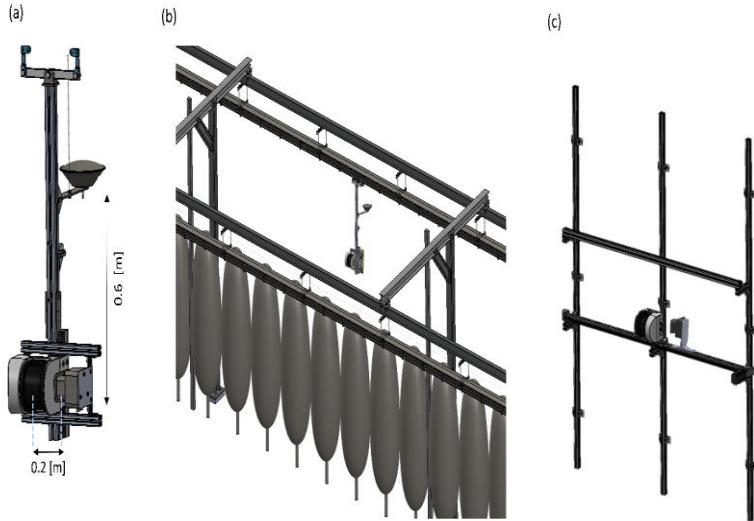


Fig.10. Representation of (a) the sensor-frame system; mounted on (b) linear tooth-belt and (c) circular conveyor system.

As described in D2.1, for defining the position and shape of apples, the geometric feature of curvature (C) was calculated for each point of each 3D tree point cloud using the k-nearest neighbors algorithm (Tsoulias et al., 2020). For this purpose, the local neighborhood of points $P_i = [x_i, y_i, z_i]$ was analysed in 3D coordinates. The total number (N) of P_i within each tree's cloud was used to estimate the mean of all nearest neighbors $P = \frac{1}{N} \sum_{i=1}^N$.

After mean centering, each P_i with P value per nearest neighbor's set and decomposition of covariance matrix. The latter was decomposed based on the singular value decomposition, producing the eigenvalues ($\lambda_1, \lambda_2, \lambda_3$), which were classified according to decreasing percentage of explained variance in the data. The eigenvalues were scaled between 0 and 100, allowing the comparison of different clusters. More specifically, the values closer to 100, the higher the likelihood for shape of points to be curved. The probability density function was performed to define the thresholds of curvature (C_{th}) and LiDAR's backscattered reflectance (R_{th}) defining the range of apple points in terms of C and backscattered reflectance (C_A and R_A). The points that fulfilled the criteria of $C_{th} \leq C_A$, and $R_{th} \leq R_A$ were segmented and categorised as apples.

This allowed to define the temperature values on the surface of apples by means of LiDAR (FST_{LiDAR}). The temperature on fruit surface was manually measured (FST_{Manual}) ($n = 285$) with an infrared thermometer (Microscanner D501, Exergen, Watertown, USA) and compared with the correspondent averaged FST_{LiDAR} . The detected apples were categorised, in west and east, based on their position on the tree side.

Evaluation

A metal tree frame with dimensions $2 \text{ m} \times 0.30 \text{ m} \times 0.05 \text{ m}$ was constructed to assess the measuring uncertainty of the phenotypic system in terms of temperature (Fig. 11). Five bars with 0.30 m distance from each other were placed horizontally on each side of a metal trunk. Sphere targets of 60 mm ($n = 3$) and 80

mm ($n = 12$) size were applied to assess the temperature derived by the phenotypic platform. The spheres were coated with white barium sulphate (BaSO_4 , CAS Number: 7727-43-7, Merck, Germany) and blackened urethane (S black, Avian Technologies, New London, NH, USA) for acquiring the minimum (S_w) and maximum (S_b) T_{LiDAR} on the sphere surface. The phenotypic system was utilised to scan the metal frame indoors and outdoors in the orchard, and an infrared thermometer to manually acquire the temperature on the sphere surface (T_{Manual}).

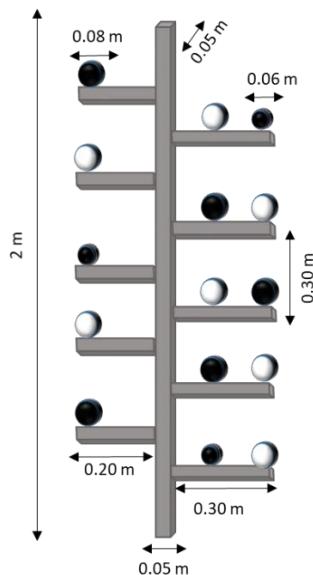


Fig. 11. Representation of the metal tree frame of known distances with sphere targets.

Descriptive statistics were applied to all datasets capturing minimum (min), maximum (max), mean, standard deviation (SD). A regression analysis was performed to quantify linear relationships between the manual measurements and the detected temperature by means of LiDAR, and RMSE, mean bias error (MBE). Descriptive statistics were carried out by Matlab (v.R2018b, Mathworks Inc., Natick, MA, USA).

The values of T_{LiDAR} on the white (S_w) and black (S_b) surfaces appeared above 19.54 °C and below 19.84, respectively (Table 1). The temperature difference between the spheres was marginally differed, since no passive heat was applied, and the ambient temperature of the room remained at 19 °C. The T_{LiDAR} was related to the T_{Manual} , revealing an adjacent coefficient of determination (R^2_{adj}) of 0.95 RMSE = 0.02 °C in SB and 0.94 with and 0.01 °C in SW. Generally, high measuring uncertainty was noticed on spheres, when the metal construction placed in field conditions (Fig. 12 b), particularly in the black coated spheres. The minimum and maximum T_{LiDAR} showed 1.5 °C difference on the surface of SB.

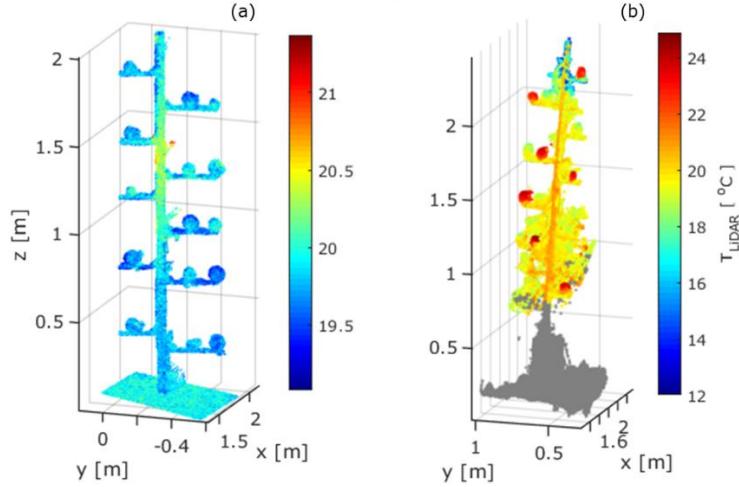


Fig. 12. The resulting thermal point cloud of the (a) metal tree indoors and (b) outdoors.

Table 1. Results of LiDAR detected temperature (T_{LiDAR}) of spheres on the metal tree indoors and outdoors ($n = 15$), regarding maximum (max) [$^{\circ}\text{C}$], minimum (min) [$^{\circ}\text{C}$], standard deviation (SD) [$^{\circ}\text{C}$], mean bias error (MBE) [$^{\circ}\text{C}$], root mean square error (RMSE) [$^{\circ}\text{C}$], and adjusted coefficient of determination (R^2_{adj}).

		min	max	mean [$^{\circ}\text{C}$]	SD	MBE	RMSE	R^2_{adj}
Indoors	S_B	19.72	19.84	19.69	0.08	0.02	0.02	0.95
	S_W	19.54	19.82	19.84	0.07	0.01	0.01	0.94
Outdoors	S_B	20.81	22.31	22.20	0.67	-0.03	0.11	0.97
	S_W	19.18	19.63	19.39	0.18	0.01	0.04	0.95

The methodology was applied in the orchard with a total number of 285 apples, 130 days after full bloom. The temperature varied in the 3D point cloud of the trees (Fig. 13 a). Tree organs, found above 2 m, revealed reduced T_{LiDAR} not exceeding 18.2 $^{\circ}\text{C}$. Moreover, the T_{LiDAR} on stem points showed a mean value of 20.6 $^{\circ}\text{C}$ with 0.65 $^{\circ}\text{C}$ standard deviation. After the application of fruit detection algorithm, the shape from 272 with an 89.7% F1 score was detected. The FST_{LiDAR} ranged between 16 and 22 $^{\circ}\text{C}$ (Fig. 13 b).

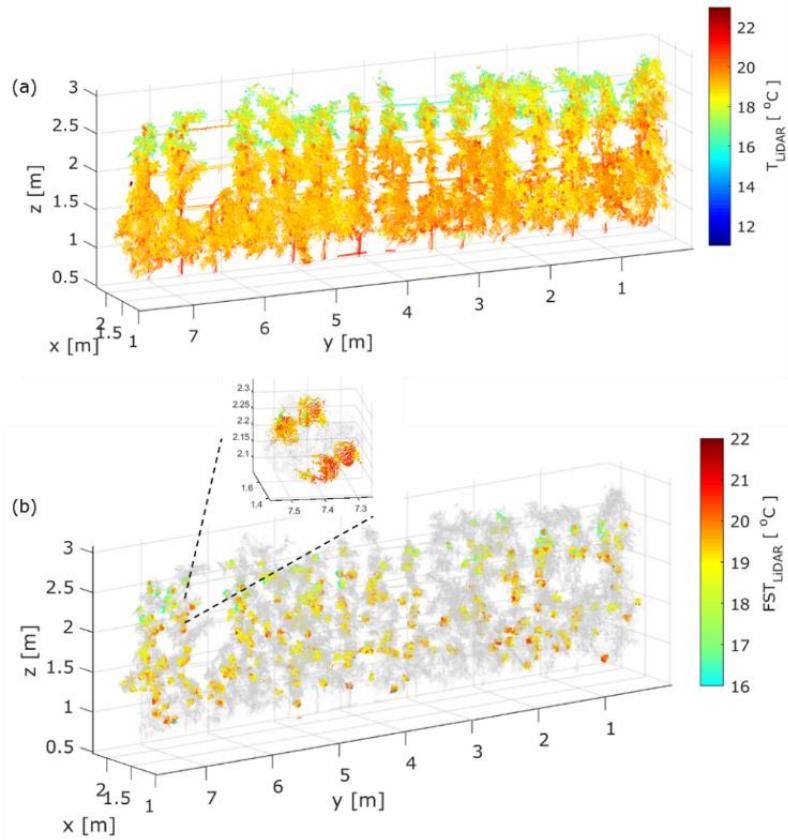


Fig. 13 Representation of (a) 3D thermal point cloud and (b) segmented temperature on fruit surface (FST_{LiDAR}) [$^{\circ}C$] in sampled trees measured with at DAFB₁₂₀.

The FST_{Manual} was related with FST_{LiDAR} of apples in the west and east side of the trees, resulting in an $R2adj$ of 0.91 and 0.99 with an RMSE of 0.25 and 0.01, respectively. The fruit located in the east side of tree developed an enhanced average FST_{LiDAR} ($18.8 \pm 0.75 ^{\circ}C$), while a less pronounced value ($18.3 \pm 0.61 ^{\circ}C$) was observed in the west side (Fig. 14). In parallel, apples from both sides depicted a similar range in terms of height.

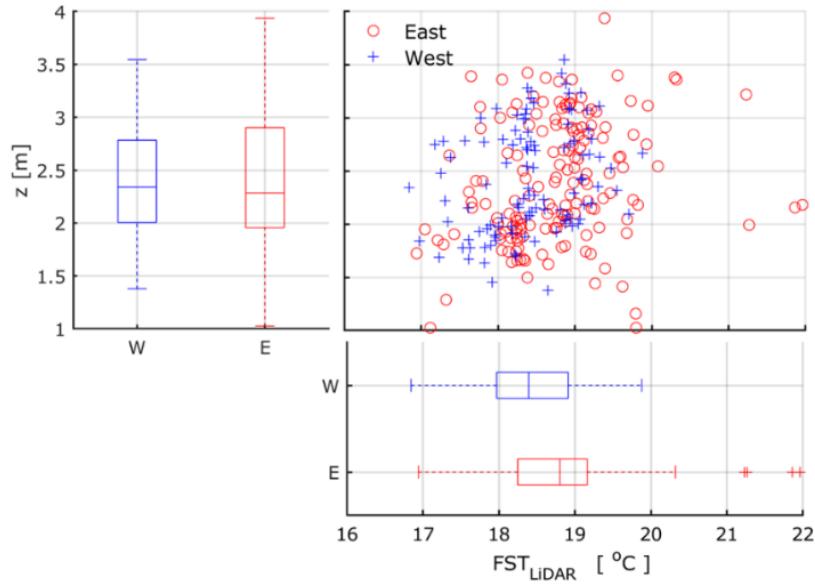


Fig. 14. Scatter plot and marginal box-Whisker plot of the segmented temperature on fruit surface (FST_{LiDAR}) and of fruit height, categorised based on the west (W) and east side (E) of the tree. The standard deviation is represented by lower and upper edges of the box, the dash in each box indicates the average.

Overall, the enhanced field of view of LiDAR laser scanner can determine the FST, which derive from the entire 3D tree profile, allowing to model fruit temperature and improve decision making in the orchard. The frequent acquisition of FST_{LiDAR} can be utilised as control measures to detect damaged fruit on the tree, increasing fruit storability and reducing food waste. The acquired FST information could be utilized, in future, for predicting various abiotic stresses (e.g. sunburn) and comprehending its effect on soluble solid content in relation with the position of the fruit in the tree canopy. The described methodology with specific customization, based on sensor availability, could be utilised for heat-stress monitoring in other perennial specialty crops.

Hypotheses were achieved

Application of the metal construction allowed the evaluation of the extrinsic calibration, presenting a highest 0.02 °C RMSE with 0.95 R²adj in the lab, and 0.11 °C RMSE with 0.97 under field conditions.

It also provided meaningful information about the FST_{LiDAR} on apples, which correlated strongly with the FST_{Manual} ($R^2_{adj} = 0.99$) in the east side of the tree.

The values of apples in the east side of tree showed enhanced FST_{LiDAR} values compared to the west side. In summary, the phenotypic system was able to detect the temperature on apple surface, a result that can be utilised in the monitoring and prevention of fruit sunburn.

Deliverable No D2.2 - Continuation

Deliverable hypothesis

Deliverable No	Hypothesis
D2.2	<ul style="list-style-type: none">• Apple fruit can be segmented from LiDAR and RGB-D 3D point cloud of trees in the orchard.• Cherry fruit can be segmented from 3D point cloud of trees in the orchard.• Grape can be segmented from 3D point cloud of trees in the orchard.

Deliverable Description

Deliverable No	Description
D2.2	<ul style="list-style-type: none">• Codes for fruit detection using RGB-D and laser scanner are available for the three fruits.

The following report describes the process and achievement obtained in relation to what anticipated in in “**Apple and grape cluster detection by means of RGB-D camera**” section of the deliverable D2.2 (pages 14-17). Here the methodology and principles utilized for the RGB-D/ thermal system (RGB-D/T) development and the further fruit positional and thermal information extraction are explained.

Sensors, platform and setup

The sensors utilized to build the RGB-D/T system are an RGB-D *IntelRealsense D435* camera (<https://www.intelrealsense.com/depth-camera-d435/>) and a *SEEK CompactPro* thermal camera (<https://shop.thermal.com/compact-pro-ff-android-usb-c>) (Fig.16). Consumer grade sensors were voluntarily utilized in the setup to investigate possible “low-budget” solutions to scan the orchards, for fruit temperature distribution, with sufficient accuracy for the purpose of the project.

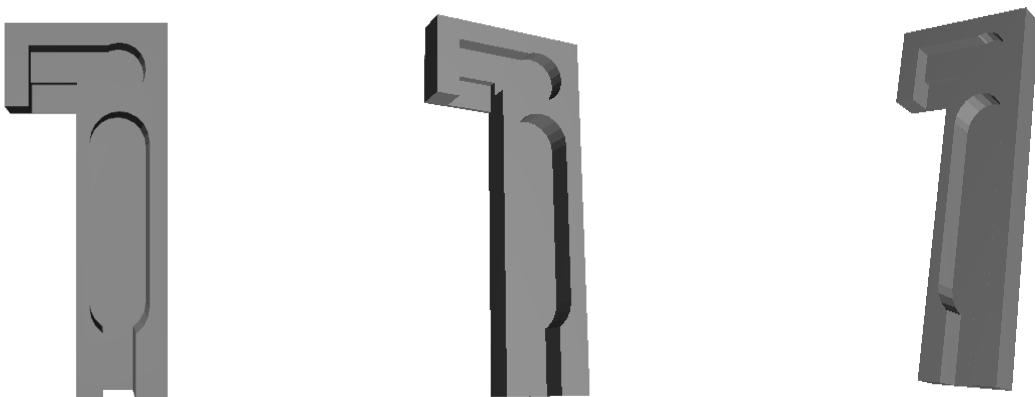


Fig.15. representation of the 3D printed case holding the thermal and RGB-D cameras

In order to firmly hold the cameras during the measurements, a 3D printed case was created and customized to fit with sensors and cables (Fig.15, Fig.16). This was designed to vertically align the center of both the

cameras lenses, while keeping the distance between them as reduced as possible, so to favor overlapping in cameras field view (FoV). Considering the cameras FoV (RGB-D camera: $69^\circ \times 42^\circ$; Thermal camera: $32^\circ \times 32^\circ$), the sensors were oriented in order to fit the trees' height with widest field of view between the sensors (i.e., the horizontal FoV of the RGB-D camera). The 3D printed case was firmly fixed (with screws) on a wood pole equipped with two bubble level and mounted on a tripod equipped as well with a bubble level.

Data collection platform consisted of a standard laptop (MSI Katana GF66), exploiting a ROS workflow. The development of a ROS workflow to collect the data was essential to synchronize the frames from the two sensors. In addition to that, while the RGB-D camera is supported with a dedicated SDK (<https://www.intelrealsense.com/sdk-2/>), the SEEK CompactPro thermal camera does not have one, featuring only an official not open source, smartphone application based on Android OS (<https://play.google.com/store/apps/details?id=com.tyriansystems.SeekThermal&hl=en&gl=US&pli=1>). Thus, ROS was used to develop a custom ROS node so that both sensors can work simultaneously from the same non Andorid-OS based platform (i.e., a laptop in our case).

The developed ROS workflow allows to generate “.bag” files containing synchronized data from both the sensors. From these files, it was later possible to extract the same timing data from the different cameras, despite their frame rates.

From now on we will referring to “RGB-D/T system” as the ensemble of both thermal cameras mounted in the 3D printed case together with the ROS workflow utilized for synchronized data collection.

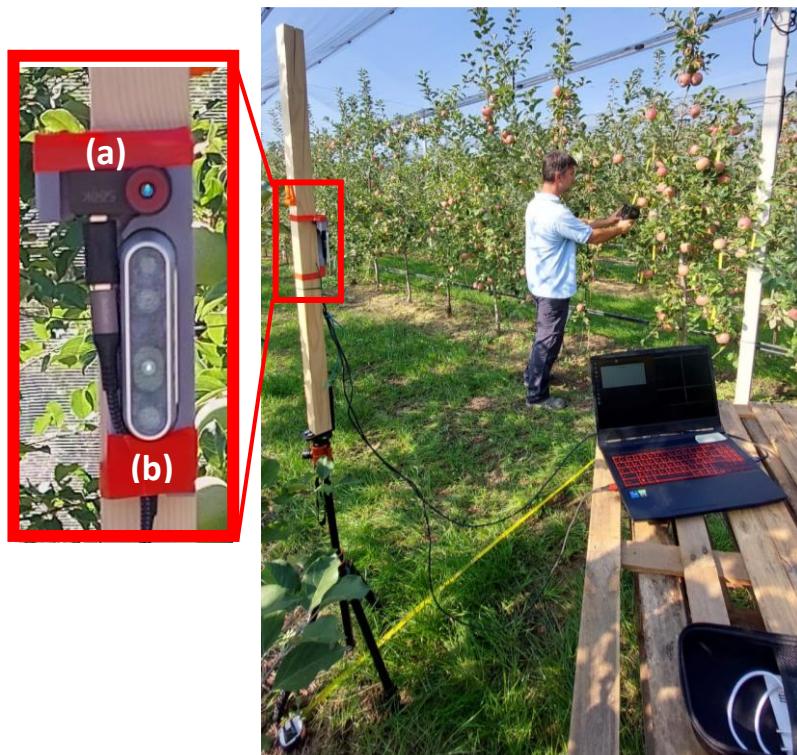


Fig.16. sensors setup: upper camera is SEEK thermal Compact Pro camera (a), while lower camera is IntelRealsense D435 (b)

RGB and Thermal image alignment

ROS-extracted images were either matrices like $1920 \times 1080 \times 3$ or $1280 \times 720 \times 3$ in uint8 datatype for RGB-D color images, while thermal ROS-extracted images are $320 \times 240 \times 1$ in int32 datatype. Considering also the different physical position from which data were collected, it was needed to align and register the two images to ensure both thermal and color information to refer to the same area/object framed in the scene.

The sensors are sensible to different wavelength range (visible vs infrared), so it was necessary to exploit both of them to proper align images. For doing that, an alignment panel was created using small lightbulbs

(n= 30, diameter 0.005m) mounted 0.125m apart on wood board following a chessboard scheme, similarly to what done by [1]. The high temperature and light emission of these bulbs, when powered, allows to distinguish the bulbs from the background both in RGB colors and temperature data (Fig. 17). From images of this panel, an alignment process was performed as follows.

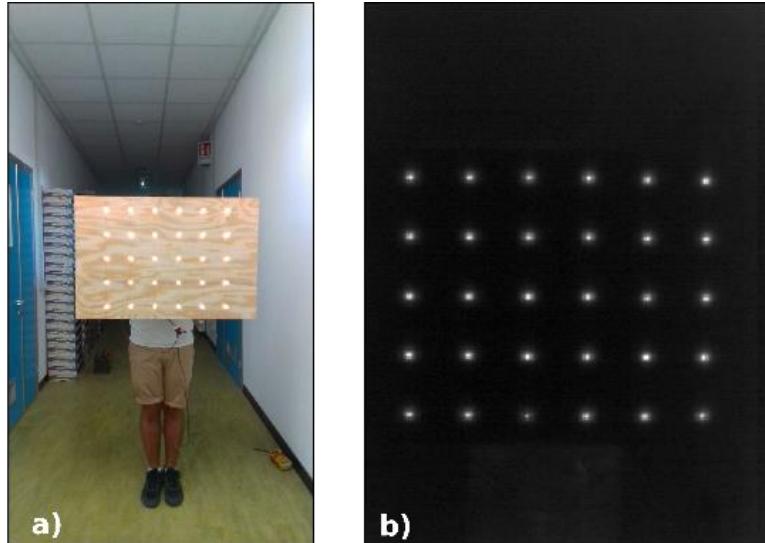


Fig. 17. alignment panel images collected at 2.6m approx. by a) RGB-D camera – res: 1920 x 1080; b) Thermal camera (normalized data) -res: 320 x 240.

1-Blob detection

To detect the light bulbs in the RGB images, a customized *SimpleBlobDetector* from opencv library (<https://opencv.org>) was exploited. Since the IntelRealsense camera may shoot images at different resolutions (i.e., 1280 x 720 and 1920 x 1080), the customization of the detector was performed to fit with the lowest resolution.

To detect the light bulbs in the thermal images, at first a matrix normalization occurred, then a customized SimpleBlobDetector designed for a 320 x 240 matrix was exploited.

The customisation concerned the following parameters (Table 2).

Table 2- Blob detectors parameters customized

CUSTOMIZED PARAMETERS	RGB (1280 x 720)	THERM (320 x 240)	NOTES
params = cv2.SimpleBlobDetector_Params()			
params.minThreshold	= 150	30	# minimum value for initialising the image thresholding
params.maxThreshold	= 255	255	
params.thresholdStep	20	20	# value to increment the thresholding value up to thresh maxThreshold
params.filterByArea	= True	True	
params.minArea	= 40	1	# pixel
params.maxArea	= 260	60	# pixels
params.blobColor	= 255	255	# search for lighter pixels
params.filterByConvexity	= False	False	
params.minCircularity	= 0.7	0.7	# square

params.maxCircularity	= 1	1	# circle
params.filterByInertia	= False	False	
params.minInertiaRatio	= 0.5	0.5	# ratio between blob axes (0: line; 1: circle)
params.maxInertiaRatio	= 1	1	# perfect circle
			# keypoints
detector= cv2.SimpleBlobDetector_create(params)			

2-Images alignment

The alignment was performed exploiting the RGB and thermal keypoints identified by their respective blob detectors. Since, keypoints are blob coordinates, these have been converted into x, y coordinates locating the detected blobs in their canvas. Then all the keypoint coordinates are filtered thanks to a blob-to-blob distance matrix, that consider the fixed distance between the lightbulbs on the panel (here represented by blobs). This to remove off-target and erroneous blobs that can be detected in case of high illuminance / reflection or other errors.

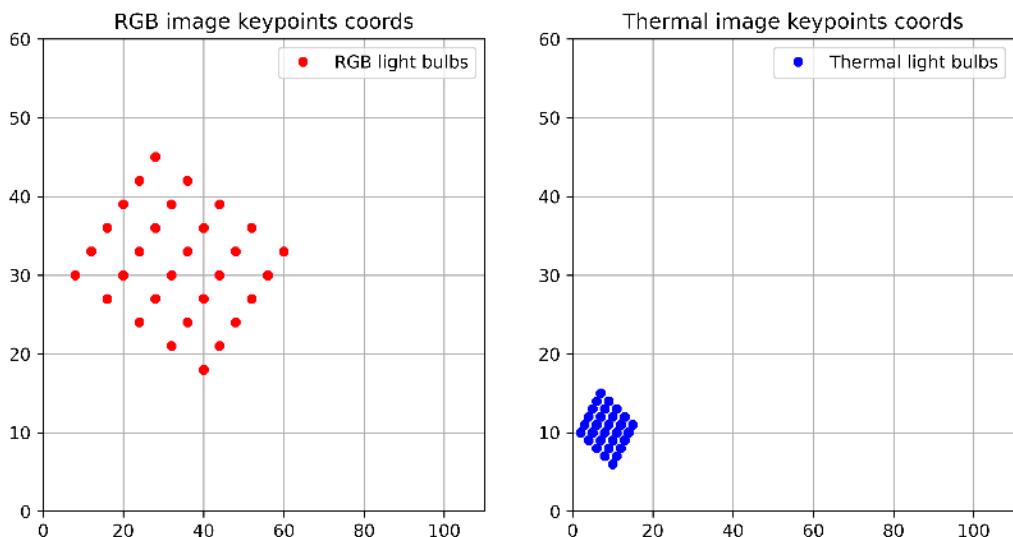


Fig. 18. Graphical representation of the difference in RGB and thermal keypoints scale for a pair ROS-extracted of synchronized images. Axis units are non-dimensional and values are intended to show differences in the RGB and thermal frame dimension scale.

Then, for all the pairs of synchronized RGB-Thermal images (both presenting 30 filtered blobs coordinates – Fig. 18), the minimum and maximum x and y values, between all the blob coordinates, are identified. From these values, four points, per each image, are created $P_1(x_{min}, y_{min})$, $P_2(x_{min}, y_{max})$, $P_3(x_{max}, y_{max})$ and $P_4(x_{max}, y_{min})$, those represent the minimum bounding box enclosing all the blobs identified (Fig. 19- a). Following, the four points related to the thermal image only are projected into the RGB canvas (Fig. 19-b) thanks to scaling factors (SF_x and SF_y) obtained as follows:

$$SF_x = (RGB_P_4_x - RGB_P_1_x) / (THERM_P_4_x - THERM_P_1_x)$$

$$SF_y = (RGB_P_2_y - RGB_P_1_y) / (THERM_P_2_y - THERM_P_1_y)$$

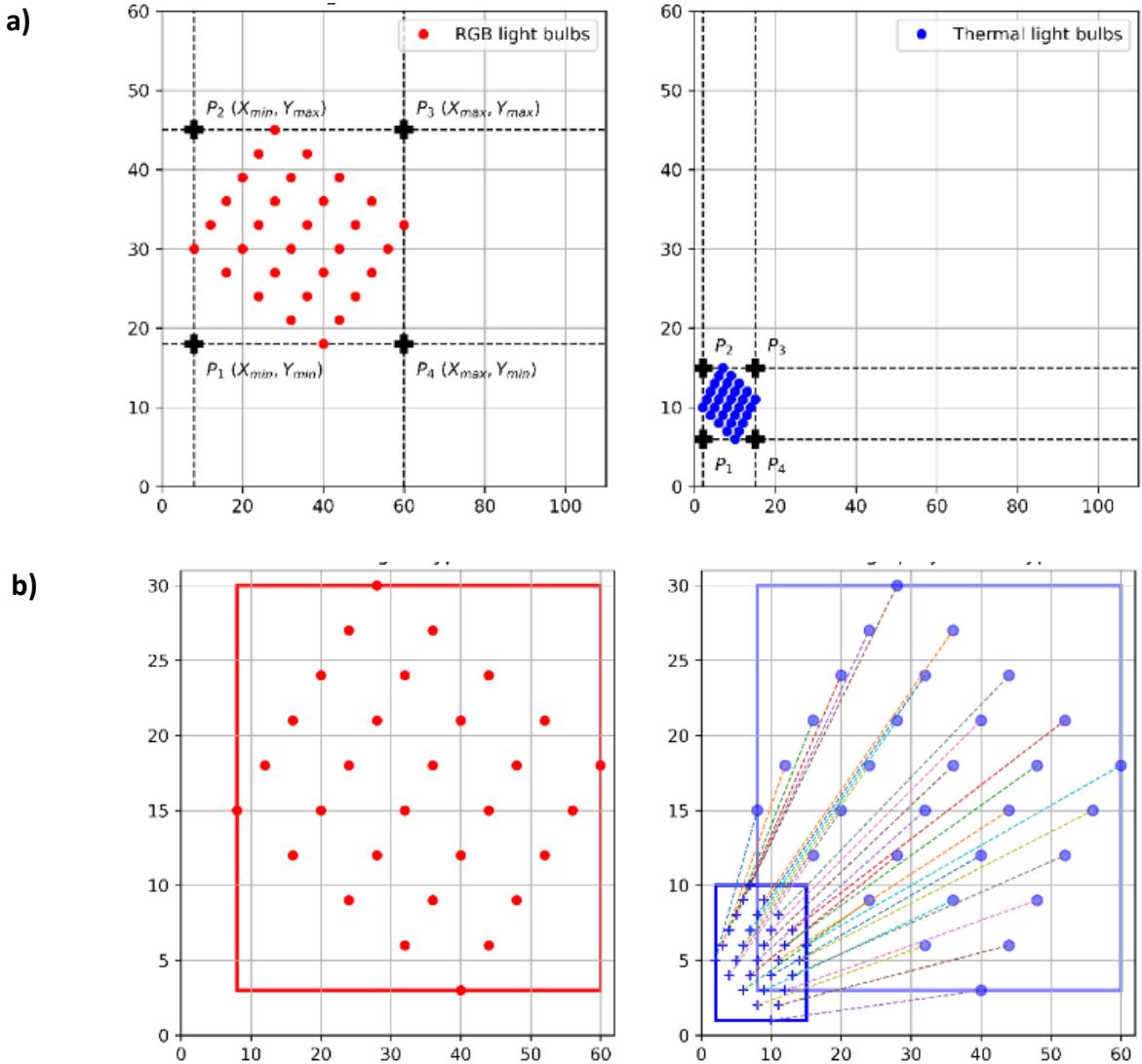


Fig. 19. a) representation of the keypoints analysis (RED = RGB, BLUE = THERMAL) to define the four alignment points (P_1 , P_2 , P_3 , P_4); b) representation of thermal keypoints' projections to match with RGB keypoints' positions. Axis units are non-dimensional and values are intended to show differences in the RGB and thermal frame dimension scale.

After the thermal-to-RGB points projection, the coordinates to align the whole thermal image so to represent the same area framed in the scene, with the same pixel dimension as the RGB reference image, are computed as follows:

```

min_x = int(P1_x_RGB - (P1_x_TERM * SFx))
max_x = int(P4_x_RGB + ((therm_img_width - P4_x_TERM) * SFx))
min_y = int(P1_y_RGB - (P1_y_TERM * SFy))
max_y = int(P2_y_RGB + ((therm_img_height - P2_y_TERM) * SFy))

```

thermal-to-RGB canva's alignment coordinated = ($min_x, max_x, min_y, max_y$)

Then, for each couple of thermal and color images used in the alignment dataset, the four “thermal-to-RGB canva's alignment coordinated” ($min_x, max_x, min_y, max_y$) are saved into a file. From this file, the average $min_x, max_x, min_y, max_y$ among all the images analyzed is computed.

Then the mean thermal image projection coordinates obtained are utilized to align the thermal image onto the RGB one (Fig. 20 Fig. 20.-a). To obtain the mean thermal image projection coordinates, 18 pairs of images

were processed from which only 6 pairs of images were correctly analyzed (i.e., with 30 blobs detected) for the alignment coordinate extraction.

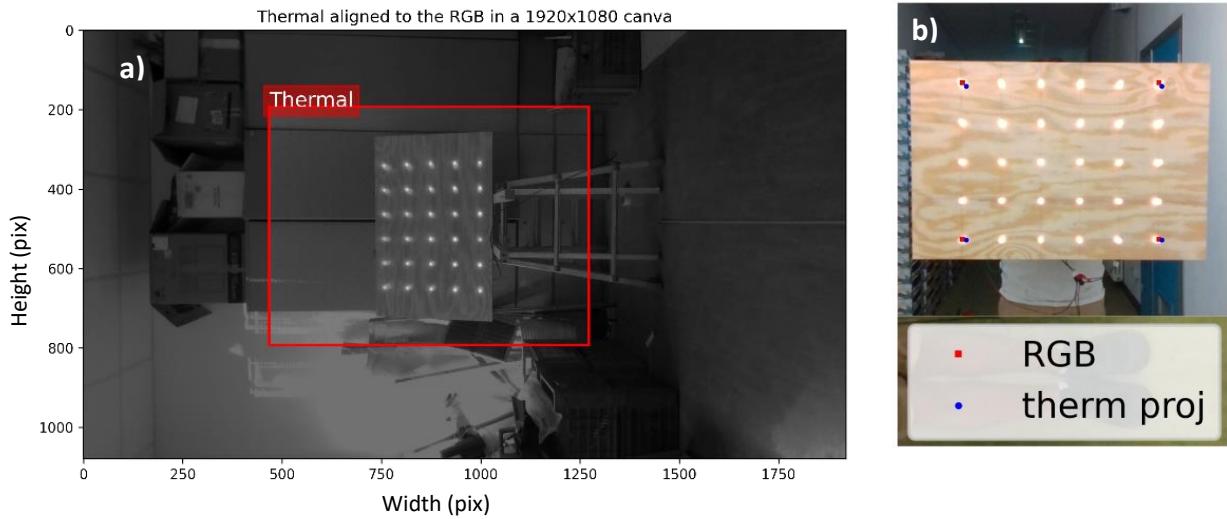


Fig. 20.. a) grey scale RGB image of the alignment panel containing the aligned thermal image at 1920 x 1080 resolution, in this case: the slight difference in lightbulbs lightness is due to the alpha merging factor between the images and the alignment errors. b) representation of the error in thermal-to-RGB pixels projection (red points - actual positions; blue points - projected positions)

Alignment evaluation

The performance assessment consisted into comparing the projected positions of P_1 , P_2 , P_3 and P_4 thermal points with the actual coordinates directly extracted from the RGB images. The evaluation pointed out a RMSE / mean error of ± 9.17 / $+4.5$ pixels and ± 4.17 / $+0.17$ pixels, on *x-axis* and *y-axis* respectively (result represented in Fig. 20-b). Considering the dimensions of target objects (apples and grape clusters), this error guarantees that most of the thermal data is related to target objects, despite inaccuracies due to alignment errors.

The alignment process just presented is needed only one time, if the 3D-printed cameras' case is used, or until the relative position between the cameras is modified. Also in case of lens focus modification the alignment process should be repeated to obtain best results. The proposed method works basically with alignment panel parallel to the cameras plane, due to the approach utilized, but this can be assumed as a minor problem, since considering the approach utilized later to extract 3D positional information, the parallelism between the cameras and the fruit tree row planes is necessary.

A suggestion for replicating the alignment process and reducing the error is so to keep the panel horizontal and stationary (avoiding movements and inclinations) at a fixed distance from the camera (2.6m approx. in our case). Additionally, shooting images in a dark environment avoiding whitish backgrounds will reduce errors during the blob-detections phase.

During the winter it is planned to improve the alignment process through a new data collection in which the encountered issues will be faced by defining optimal blob detection parameters in relation to object distances. This will allow to automate and adapt the alignment process to different distance (and or resolution). A trial for dark scene data collection will be carried out to test possible improvements.

Field data collection with the RGB-D/T system

The creation of the sensors' platform together with the development of the ROS node and workflow, and the images alignment process resulted time consuming, but necessary, to make possible field data collection. The entire system resulted properly working, for data collection, only around August 2022, so after the harvest of apples monitored in the trial. The first field data collection useful to test and evaluate the developed system was done during August-September 2022.

Field data collection consisted in a brief video recording (3 seconds approx.) of a single tree, through the ROS workflow from both the cameras. In base of the specie (apple, grape) a different recording protocol was utilized, due to the different plant training system.

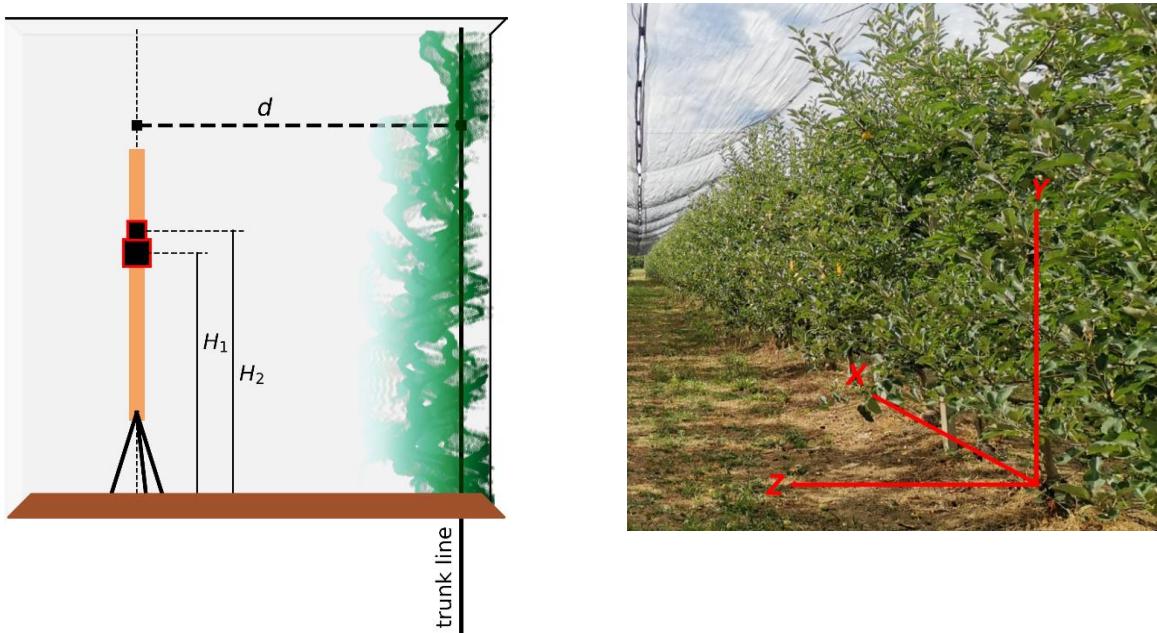


Fig. 21. Cameras positioning during the field data collection. Cameras are placed parallel to the tree row plane at a "d" distance. Field data collection height was measured in the middle point between H1 and H2 (i.e., center camera height from the ground).

Apple data collection

For apple trees, trained as "thin" spindle, the tripod was positioned in front of the tree trunk at 2.80 m distance (*Z dimension*, perpendicular to the tree-row plane - Fig. 21), with cameras parallel to the tree-row plane (*X dimension*). Considering the possible tree height, after the tripod positioning, two height recordings occurred: one at 1.40 m (*h1*) and one at 2.50 m (*h2*) from the ground (*Y dimension*, elevation from the ground - Fig. 21). The system was levelled thanks to the three level bubbles present.

The system positioning and height*distance were set according to the lowest camera FoV (thermal camera: 32°) so to include in the scene as much part of the tree as possible, while not reducing excessively the real object pixel resolution (i.e., actual mm/pixel). With the presented distance*height, the area framed by the thermal camera was of 1.55 x 1.55 meters, with 5.6mm/pixel as object resolution. Since the tripod was positioned in front of the trunk, the area framed in respect of the trunk was of ± 0.77 m, along *X* dimension, considering the trunk in the middle (*X/2*) of the framed scene. These dimensions were enough to frame one entire tree in width and having a minimum reliable analysis resolution of 11.2 x 11.2 mm (i.e., analysis of a 2 x 2 pixels matrix). Considering camera's height, *h1* framed approx. from 0.60 m to 2.20 m from the ground, while *h2* framed from 1.70 m to 3.30 m from the ground.

With this approach data were collected on 6 trees presenting red fruit (cv Gala) and 6 trees presenting green fruit (cv Fuji) for a total of 24 tree recordings (12 trees* 2 heights).

Initial idea of the proposed protocol was to exploit the *X/2* trunk position, to ease the positional information extraction, while the two-height recording were taken to later investigate the possibility to merge the collected images, obtaining information on the whole tree fruit distribution in one step, compared to multiple height recording analysis.

Grape data collection

For grape data collection, a similar set up using the same *X*, *Y*, *Z* dimension and approach was utilized (Fig. 21). In this case the tripod was placed in front of the middle of the plant canopy, and not the trunk, at 2.30

m distance approx.. resulting in a framed area of 1.32 x 1.32 m approx., with 4.8mm/pixel as object resolution. Camera's height was 1.20 m from the ground, framing the scene from 0.60 m to 1.90 m. In this case the dimensions were enough to frame one entire vine in width and height, having a minimum reliable analysis resolution of 9.6 x 9.6 mm (i.e., 2 x 2 pixels matrix).

In this case, the tripod positioning, made the trunk to be framed near to the right limit of the image collected (X_{max}). The initial idea was to exploit the X_{max} trunk position, after a manual cutting of the collected image, so to ease the positional information extraction.

With this approach, 16 single vine recordings were collected, presenting different level of defoliation / fruit occlusion.

Fruit Detection

Once RGB and thermal images collected can be aligned, it is possible to exploit the developed fruit detection models (see deliverable D2.2 – “Apple and grape cluster detection by means of RGB-D camera” section; pp 14-18), so to identify and locate fruit on the RGB image and, later, utilize the RGB detection coordinates to apply thermal image analysis on the correct area of the aligned thermal image in order to refer to the same detected object in the scene (i.e., the fruit). The trained models developed were two YOLOv5 object detection models (one for apple and one for grape, were trained reaching performances of $mAP = 0.734$ and $F1\text{-score} = 0.74$ ($cf = 0.378$) for apple (Fig. 8), and $mAP = 0.973$ and $F1\text{-score} = 0.96$ ($cf = 0.57$) (Fig. 9). These were applied on the ROS- extracted RGB images, collected with the RGB-D/T system.

In preliminary evaluation they were observed two main issues: i) on apple side, many highly occluded fruit were detected, while on grape side ii), grape cluster were undetected in most of the cases. Regarding i) the issue came out during the thermal data analysis, in fact, it was found that working on highly occluded fruit could alter thermal analysis; this because of the image alignment error, that make some non-fruit areas included in the fruit thermal analysis, and also because highly occluded fruit would not have well represented target fruit of the study (i.e., sunburn-susceptible fruit, with high probability of sunburn damage). Regarding ii) the low detection rate itself was the issue, and the main cause was found in the wide different cluster dimension in the images used for the training (larger) compared to the one collected in field from the RGB-D/T system (smaller). To overcome these two issues, a second round of training was implemented: for i) it was adjusted the labelling for apple detection models in order to detect only highly visible fruit (occlusion rate < 30%) discarding the others; for ii) the original image utilized for the training was compared to images collected in field and then modified applying a “zoom-out” alteration, so to have similar cluster dimension to the target one.

Model training – 2nd round

Considering the unexpected low performance rate occurred before the second round of training, in this, it was decided to move to a larger model working at higher resolution (i.e., yolov5l6 – res 1280 pixel), compared to the one used previously (i.e., yolov5m – res 640 pixel), so to improve performances. In add to that, it was decided to split all the images of the original dataset in train (80%) and validation (20%) sets, while avoid creating a test-set using these images. For the testing it was then decided to use directly images collected with the RGB-D/T system in field. This was done to increase the training images number and test on more real-world scenario the performances.

Furthermore, it was decided to increase the rate of added-noise images, during the augmentation process, considering the lower resolution and higher noise presented by the color images collected in field with the RGB-D/T system. For this second round, the training was performed for 500 epochs on the same workstation utilized in the previous training round.

Model Performances –2nd round (preliminary)

Due to the time needed for the accurate labelling of the test set, at the moment is not possible to show metrics regarding the performances on images collected directly in field.

Despite that, it can be said that the main issues aforementioned were faced, since with the latest models' version the obtained results were satisfactory and permitted to go forward with the system development and testing (Fig. 27, Fig. 34).

Trunk detection

Despite what previously presented in relation to 3D coordinate extraction (Field data collection chapter), it was decided to shift the approach of defining the trunk position (i.e, the origin for the Y, X, Z fruit coordinates extraction) in the images from exploiting its hypothetical position due to the camera distance*height*positioning, to actively detecting it in the image. This to increase the accuracy of the whole system, automating it, and avoiding the need of manually crop / cut the images. For doing that, it was started the development of a trunk detection model (currently on going).

Similarly to what did for fruit detection model creation, data acquisition consisted in the creation of two datasets (one per each fruit species) on which to train a CNN algorithm to detect apple and vine tree trunks (i.e., objects). Due to the difficulties encountered in the fruit detection models, it was decided to create the needed dataset. For doing that, 200 images approx. for each specie were collected with different cameras sensors (Smartphones: Asus Zenphone 5z, Huawei P20; Others: Intel Realsense D435i). At current time, the dataset is still under labelling, so no preliminary results can be presented for this model.

The approach for the training phase will be the same utilized for the 2nd round of training just exposed: image augmentation favoring target object dimension and characteristics with dataset splitting in train and validation set (80%-20%), a test set based on images collected in field through the RGB-D/T system, a yolov5l6 object detection model, 500 epochs of training on the same workstation.

Fruit temperature extraction from RGB-D/T-system collected images

In this section will be explained the steps occurred in order to extract temperature information from the fruit detected on the RGB images collected in field through the RGB-D/T system .

1-Thermal camera raw to °Celsius thermal data conversion definition

The thermal camera utilized is a consumer grade camera build for Android smartphone ; it has a mobile application (Seek Thermal) in which the proper computation and conversion are applied so to present the raw thermal data obtained by the sensors, in to Celsius/ Fahrenheit degrees. Considering that data collection was done through a ROS workflow, due to the cameras' synchronization requirements, it was not possible to obtain processed thermal data through the utilization of the official application. Since the thermal conversion equation used by the official application, is neither opensource nor available for developers, reverse engineering approach was exploited to get the proper calibration functions and convert raw thermal data in Celsius degrees.

For doing that, 6 thermal images (resolution: 320 x 240 pixels; 76800pixels in total) of known temperature target objects (lightbulbs or steel bottle with hot water and cold refrigerated container) were collected at different distances (one per each distance of 0.5 m, 1.0 m, 1.5 m, 2.0 m, 2.5 m, 3.0 m) with the official Android OS SEEK application in lab condition. The Seek official app allowed to collect, for each distance, all the thermal data of the scene, storing it into a “.tiff” file composed of three layers (or channels) containing thermal data respectively in color-mapped, Celsius degrees and raw data format. Thus, for each image (i.e., distance) pixel

values of Celsius degrees and raw data layers were analyzed to extract a general regression function between them.

During the data analysis, it was found that the relation between the raw data and Celsius degrees layers, is not completely linear. As shown in Fig. 22 the mean relationship between these layers, among all the six collected images, change in base of the raw thermal domains (i.e., object temperatures ranges): Fig. 22-a shows three clear raw thermal data domains in which relationship change (<3000 , $3000-7000$, >7000), Fig. 22-b shows the detail of the domain $3000-7000$, where the average linear regression seems to fit good with the presented thermal domain.

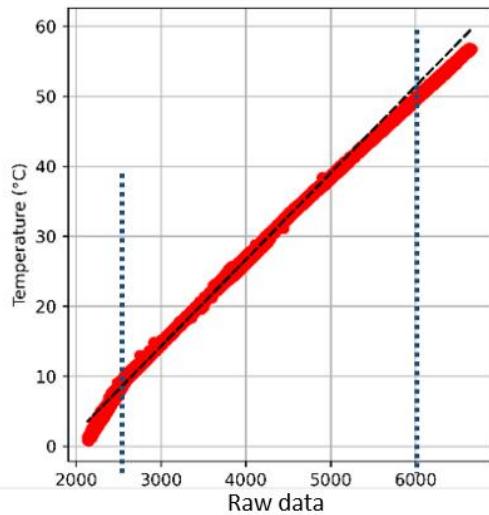
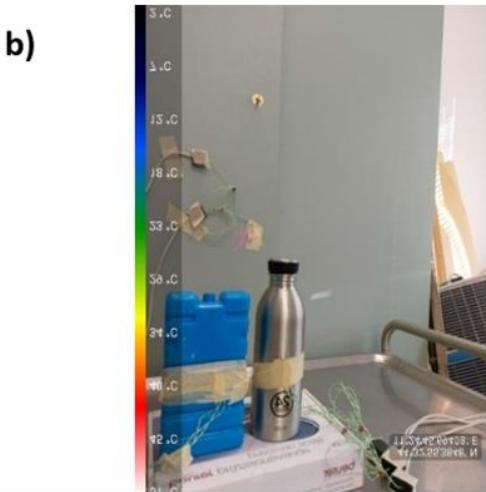
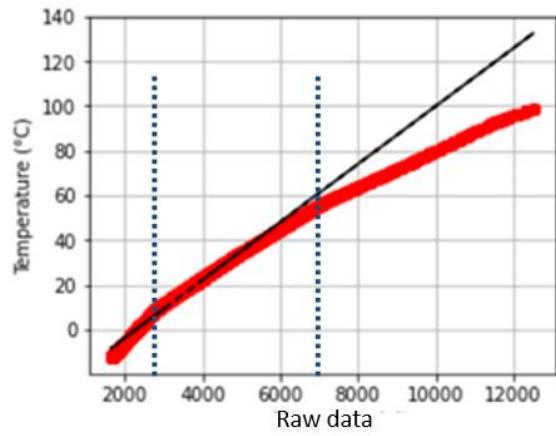
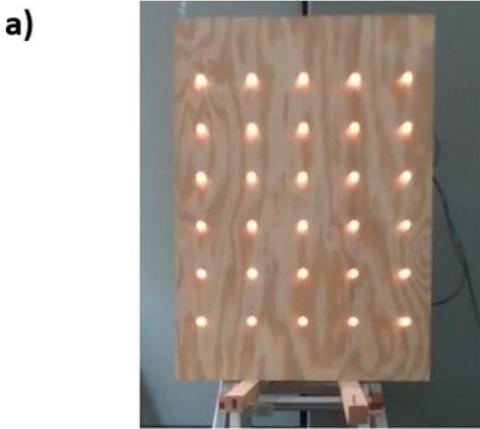


Fig. 22. Thermal calibration set up (left) and related plots of raw thermal vs Celsius degrees obtained through the official SEEK smartphone (right); red dots represent the plotted data, black line is the general linear regression. A) Thermal calibration set up exploiting the alignment panel with powered, hot lightbulbs; b) Thermal calibration set up exploiting known temperatures object: steel bottle-hot, refrigerated container-cold, background-ambient; detail on the good fitting of the linear regression in this temperature domain). (N=76800 * 6)

Following, it was analyzed the effect of the distance and the object size on the thermal data collection. In this case, not all pixels in the raw data and Celsius degrees layers were compared (as in Fig. 22), but only those related to minimum, ambient and maximum temperature that were represented by fixed temperature and fixed position objects inside the scene. A manual image segmentation of these objects was done to extract the related thermal data.

In Fig. 23-a is shown how the extracted thermal data changes in base of distance: maximum temperature decreases rapidly with increasing distance for small objects such as light bulbs (5 mm diameter), while ambient and minimum temperature increase with a slower rate (objects size $> 0.1 \times 0.1$ m: refrigerated container, background wall area). On the other hand, in Fig. 23-b, the temperatures evaluated result more stable with increasing distance, also for maximum temperature. This is mainly due to the bigger size of the target object (hot steel bottles for maximum temperature, same cold refrigerated container, and background

wall area for ambient and minimum temperature). In Fig. 24 are shown the temperature errors related to distance: it is clear how the error range at 0.5-3.0m results higher for small target objects (lightbulbs) compared to bigger ones (“steel bottle”).

Both distance and the object size resulted to affect the thermal accuracy since the amount of thermal radiation perceived by the camera decreases with distance, and smaller objects emit less thermal radiation than larger objects. Thus, in add to the raw data-to- Celsius degrees conversion equation, a distance – based correction appears necessary to obtain reliable data when collecting fruit temperature in field (i.e., with distance > 2m).

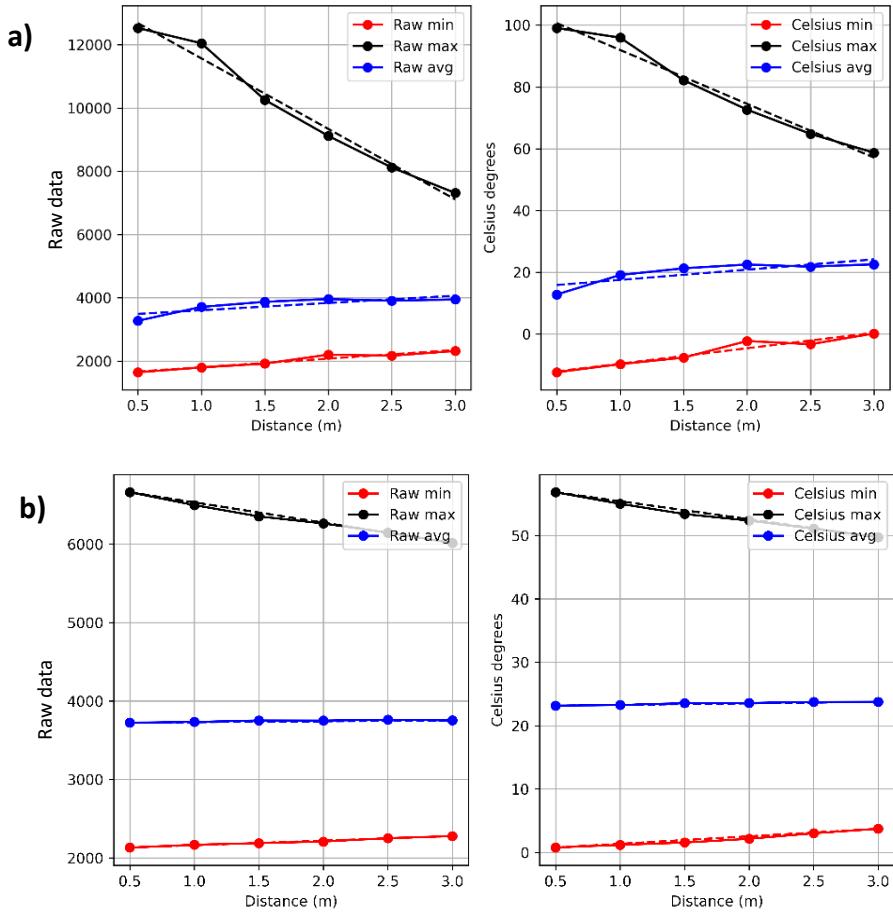


Fig. 23. Effect of distance on the temperature estimation of the same object. From left to right are shown raw thermal data and Celsius degree data for ‘max’, ‘avg’ and ‘min’ that are respectively maximum, ambient and minimum temperatures in the scene; these are represented respectively in a) by small lightbulbs, background wall area, refrigerated container; in b) by steel bottle, background wall area, refrigerated container.

Currently, a proper distance-based correction procedure (based on more than 6 discrete points distant 0.5m apart) is still under development; but with the data available, it was opted to extract a series of optimal raw-to-Celsius conversion function considering together temperature domain and distance. So, per each temperature domain (<3000, 3000-7000, >7000), six optimal linear function coefficients (a , b) for applying this conversion were extracted (1 per each distance check point). Since the thermal calibration set up using the lightbulbs presents a wider temperature range, which already includes the one investigated with the “steel bottle” dataset (Fig. 22-a and b), it was opted to utilize the optimal linear functions coefficient obtained by this set up (Table 3).

Table 3. Optimal raw-to-Celsius linear function coefficients in base of distance and raw thermal data domain. Data are obtained utilizing the alignment panel with powered, hot lightbulbs. (N= 76800)

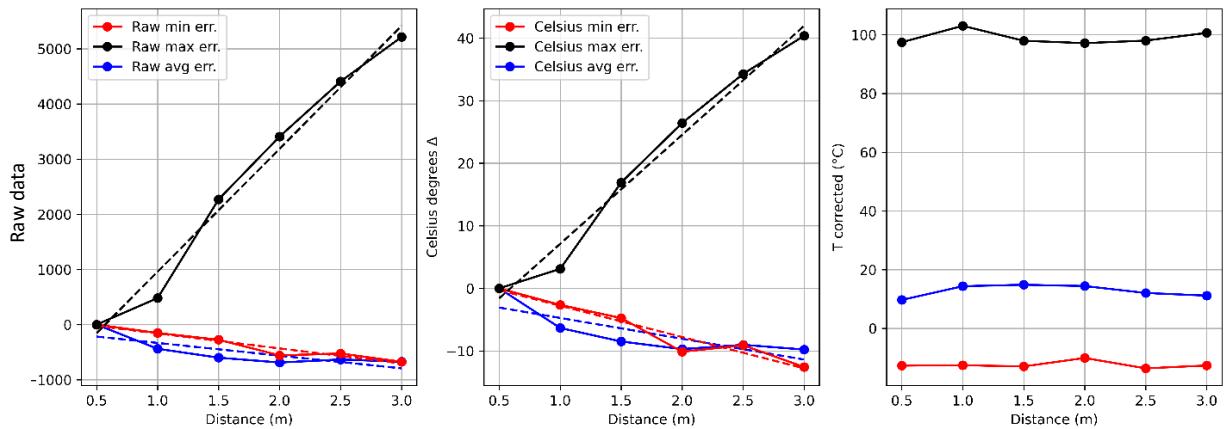
	distance	Slope(a coeff.)	Intercept (b coeff.)
General			
	0.5m -3.0m	0.01286527470000000	-28.628279400000000000
Raw <3000	0.5m	0.01790580184650830	-41.62511947512870000
	1.0m	0.01709801137150510	-39.87904853972370000
	1.5m	0.01671930703052060	-38.82596994946430000
	2.0m	0.01648162214344660	-37.84663166550710000
	2.5m	0.01636317146858520	-37.77315677927730000
	3.0m	0.01592340420569740	-36.11466921652480000
Raw 3000-7000	0.5m	0.01122734467909650	-22.07632178280650000
	1.0m	0.01121187326341270	-22.00337506467050000
	1.5m	0.01112760151137210	-21.64788385102760000
	2.0m	0.01112956235348940	-21.55103524359960000
	2.5m	0.01106067709941020	-21.37977980029620000
	3.0m	0.01111150330406360	-21.29391749734170000
Raw >7000	0.5m	0.00802669863747835	-0.37960504788257500
	1.0m	0.00810378210333579	-1.07049399213111000
	1.5m	0.00794082388563919	0.21424855976091400
	2.0m	0.00785087065611661	1.01781066231241000
	2.5m	0.00785149354366968	0.93286503219869300
	3.0m	0.00400799716597930	29.31449127197260000

The effect of object size was not considered in this conversion equation since during the field utilization, fruit can be assumed with a dimension of the same magnitude, not presenting a large size difference as the one for the objects analyzed in the two thermal calibration set up (lightbulbs vs steel bottle).

2-Preliminary temperature-to-distance correction

Despite what presented above, it was tried to extract a preliminary temperature-to-distance correction to improve thermal estimation results. Data shown in Fig. 24.1 24 represent the temperature error, at each distance, for the thermal calibration set up previously presented. Despite the different error range between lightbulbs (40°C) and steel bottle (7°C) set up collected data, it can be seen how the slope of the linear models fitting the maximum temperature in the two datasets result quite similar. Using this error data, a linear regression equation was computed to account for the temperature variation in relation to distance (Table 4).

A)



b)

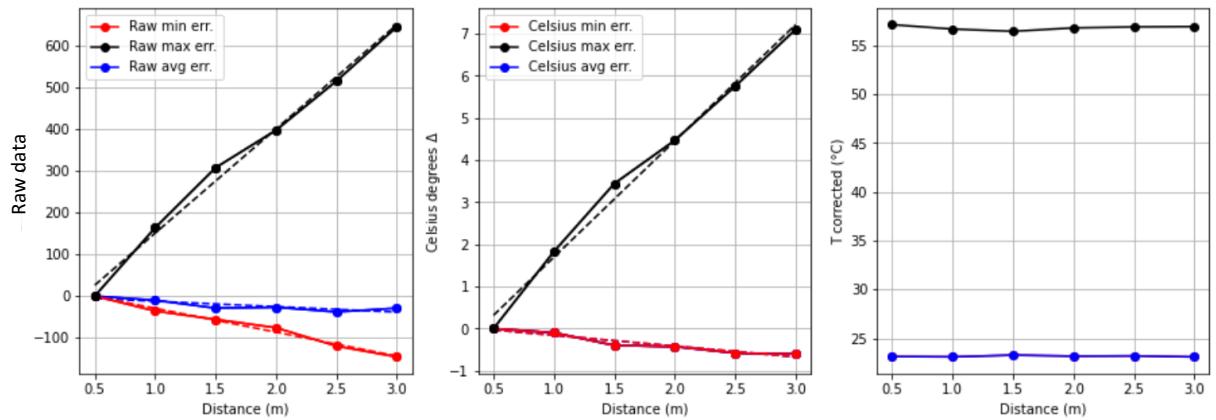


Fig. 24.1. Temperatures errors related to distance of collection in raw (left) and °Celsius format (central); dashed lines represent linear regression through the points. On the right, the temperature values after the application of the correction equation. a) refer to lightbulbs set up; b) refer to steel bottle set up.

Correction coefficients were computed for each of temperature (minimum, ambient and maximum) since their error behavior, in relation to distance, is different. In Table 4 are reported the linear coefficients of the preliminary temperature distance correction functions extracted by the interpolation of the pictures taken at the 6 distances for the steady temperature object in the scene.

Table 4 – Preliminary linear function coefficient for temperature to distance correction

	Slope(a coeff.)	Intercept (b coeff.)
Min_temp	-1.19148175	0.78746252
Max_temp	2.75930525	-1.06405436
Avg_temp	-0.25654206	0.09853837

Despite this preliminary approach to extract a temperature correction equation, a more robust data collection will be carried out in the future. The method hypothesized for doing that is based on a continuous temperature measurement of objects, with dimension similar the ones of the fruit in field (i.e., 25-150 cm²), presenting a steady temperature in the range of interest (i.e., ambient to 60-70°C). In this way we hope to extract a more robust distance-based calibration function.

3-RGB-D/T-system extracted thermal data conversion to SEEK app range

After the analysis for the extraction of a raw-to-Celsius conversion equation, based on SEEK app collected data, thermal data obtained through the ROS workflow were analyzed. For doing that RGB-D/T system was placed in field and the same scene was collected through the official SEEK app and the ROS developed workflow in a reduced timeframe (< 5min), from the same stationary position. During the data analysis it came out that raw thermal data collected from the different software resulted highly different, in their scale range, as can be seen in Fig. 25Fig. 25 below.

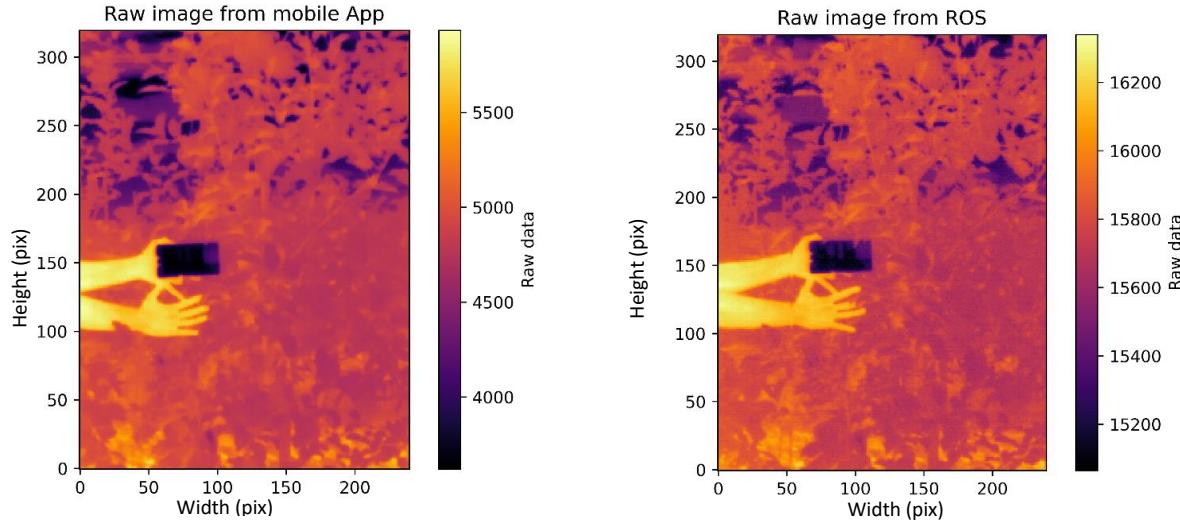


Fig. 25. Comparison of scale values between SEEK app and ROS -collected raw thermal data for the same identical scene

Because of that an adding conversion step, from ROS raw thermal data range to SEEK app raw thermal data range, was needed in order to be able to apply the previously extracted equations for temperature estimation (in Celsius degrees – Table 3). Different approaches were evaluated to do so: i) computing a converting coefficient for different temperature ranges (i.e., percentile “equalization”), ii) utilizing a remapping function based on min and max values of the images ranges, iii) applying a pixel-to-pixel regression analysis. Of the tested approaches i) was discarded due to the occurring data “polarization” inducing a high error and a reduction in data resolution; ii) was discarded as well, since the remapping function results are valid only for same minimum and maximum temperature in further collected scene, but this is not the case when moving to field data collection. Methods iii) was then considered the best approach, despite its correlation coefficient and RMSE ($r = 0.6023$; RMSE = 251 raw unit), since it maintained the correct data resolution and representation (Fig. 26), while narrowing considerably the range of ROS extracted raw thermal data to the one related to the app environment. Coefficients of the extracted ROS-to-SEEK app raw thermal value range are reported in Table 5 below.

Table 5 - ROS-to-SEEK app raw thermal value range conversion coefficient.

	Slope (a coeff.)	Intercept (b coeff.)
ROS_to_APP function coefficients	1.16550762	-13529.4453

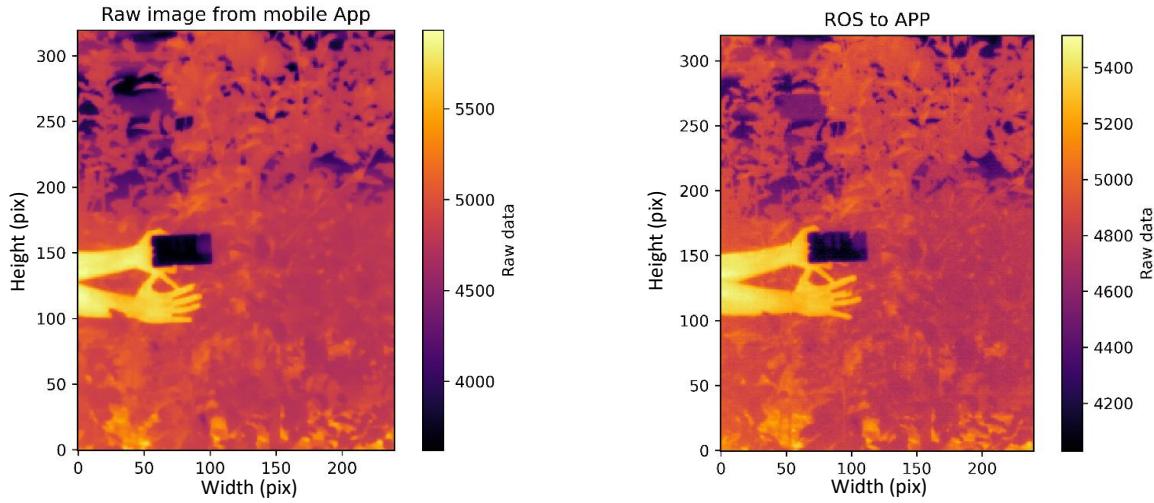


Fig. 26. Original SEEK app data (right) and result of the ROS_to_APP function application to ROS-collected thermal data (right). Is highlighted the reduced, but proper scale range, and the good representation of the thermal variability after the conversion.

Fruit temperature information extraction process

Once defined which conversion equation utilizes to obtain temperature data in Celsius degrees, from raw data collected by the thermal camera, is then possible to start working on fruit temperature data extraction. The following step are undertaken for reach this goal.

- 1- As anticipated, the fruit temperature extraction starts using the YOLOv5 object detection algorithm trained for fruit detection. The algorithm defines the possible fruit (detected) from which to extract temperature information. Per each image analyzed, is created a “.txt” file in which are stored the class name, the center coordinate of the detected fruit, the width and the height of the bounding box (bbox) of all the detected fruit.



Fig. 27. YOLOv5 trained model application for fruit detection: fruit detected are represented by red rectangle (bbox)

- 2- Exploiting the previously presented approach, the raw thermal image is aligned to the RGB one and then is clipped at each fruit bbox coordinate present in the YOLOv5 “.txt” generated file, creating an aligned thermal bbox (Tbbox) containing raw thermal information of the fruit (Fig. 28).

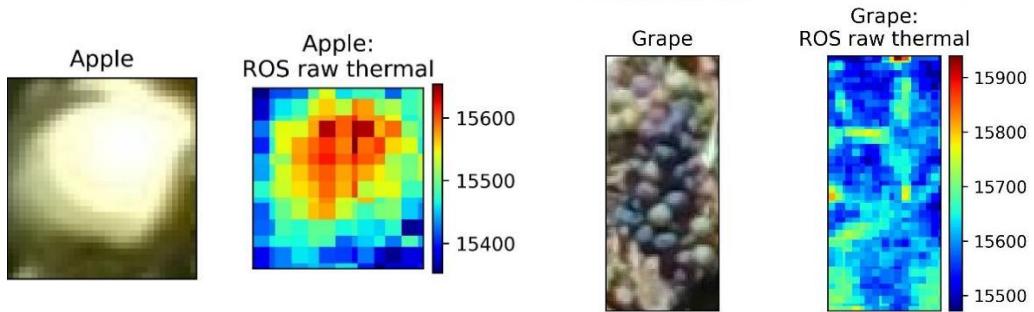


Fig. 28.. Examples of RGB (left) and relative aligned ROS-raw thermal data (right) bounding boxes

- 3- Per each clipped Tbbox, a filtering step is applied to check if the detected fruit (on the RGB image) falls in the thermal camera collecting area. For doing that an “*if / else*” statement, based on mean raw temperature value of the Tbbox was utilized: *if* the mean raw thermal data value of the Tbbox is above 14500, the process continues to temperature calculation and extraction, *else* the detected fruit is passed and not analyzed. The threshold value of 14500 was found after a testing phase that reported what presented in Fig. 29: below this threshold, approx. 40% of the fruit bbox was not falling in the thermal data collection area (i.e., 40% of thermal data = 0), presenting so not reliable data from which to extract representative fruit temperature information.

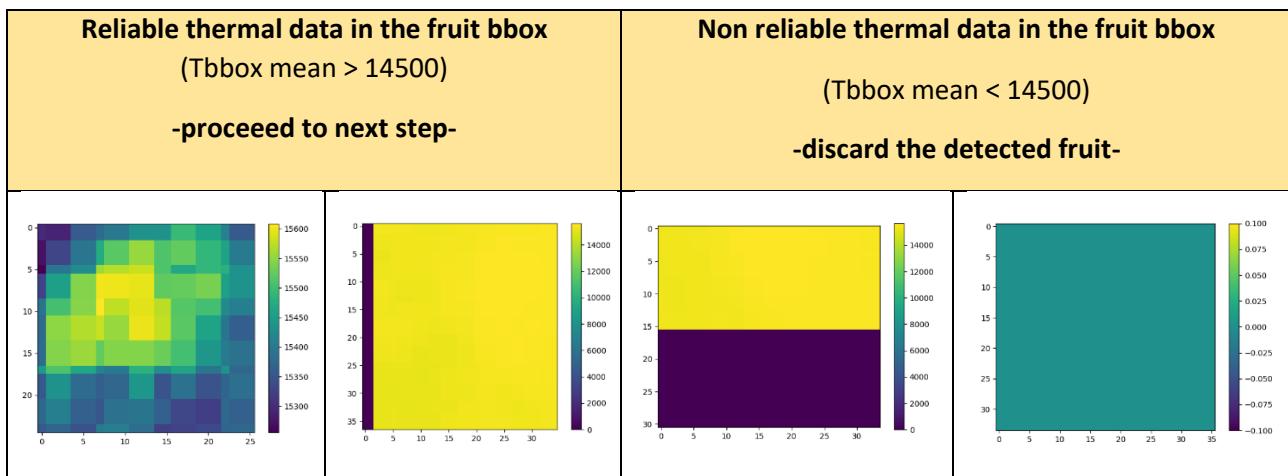


Fig. 29. Representation of filtering step for defining reliable/non reliable thermal data contained in a detected fruit bbox.

- 4- Considering the purpose of the system, the most interesting fruit temperature data are those related to the highest values, considering that the hottest fruit areas are the one which potentially can encounter sunburn damages. Based on that, on the maintained Tbbox, a percentile filtering step was developed so to extract the temperature of the warmer spot/areas of the fruit/ bbox (Fig. 30). In this step, all the pixel values below the 70th percentile were discarded, and not considered in the further steps. This filtering is essential also for removing thermal information not pertinent to the fruit, but included in the fruit detection bbox area (i.e., background, small areas with overlapping leaf, etc.).

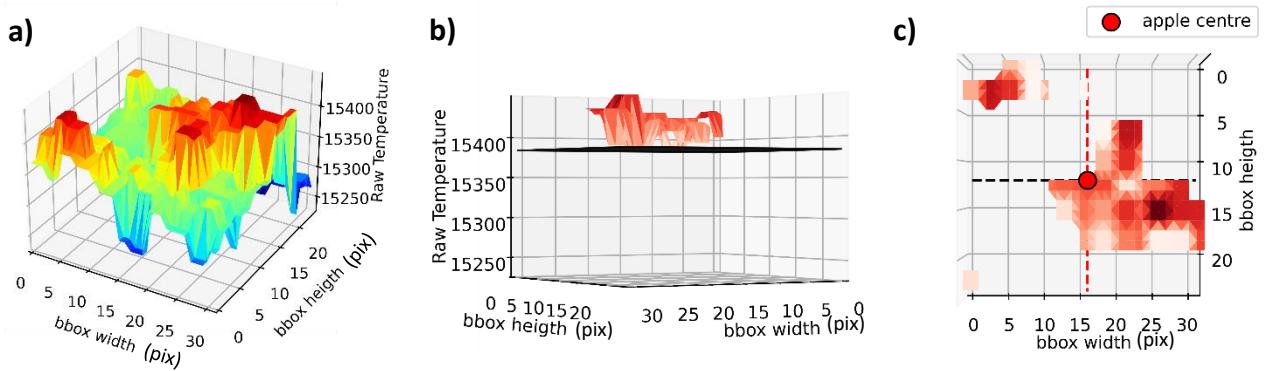


Fig. 30. a) raw data thermal profile of the Tbbox; b) quantile filtering removing data <70th perc.; c) 70th perc. filtering result

- 5- After this quantile filtering step only raw thermal data of the 70th – 100th percentile remain for the following conversion calculation and extraction:
- first raw thermal data are converted in their official seek app values range using the previously presented conversion function (Table 5);
 - then according to the obtained value and shooting distance the optimal raw-to-Celsius conversion function is applied (Table 3).
 - At last, the minimum, mean and maximum temperature of the spot are computed and corrected for the distance error (Fig. 24). Since in most of the cases fruit temperature will be always near or above ambient temperature, these temperatures are corrected using only max and avg temperature related correction coefficients presented in Table 4.
 - Fruit temperature information are then saved in a csv dataframe.

Evaluation

Errors in the fruit temperature evaluation can be due to inaccuracies in the alignment of thermal and RGB canvas. See the paragraph above for details related to this error.

The evaluation of the temperature estimation performances was made throughout a comparison with a factory calibrated semi-professional handled thermal camera (HTI-HT A9; <https://hti-instrument.com/products/ht-a9-thermal-imager>). In Fig. 31 are shown the results of the same data collection both for HTI and SEEK thermal camera in the laboratory thermal calibration set up based on lightbulbs. It can be seen how the SEEK camera tend to present higher maximum temperature and lower minimum temperature than the HTI, with an increasing error up to 1.0-1.5m, that remain then stable. The average error among all the distance, for maximum temperature resulted of +20.3°C, while for minimum temperature resulted of -4.7°C. In this case ambient temperature was not monitored.

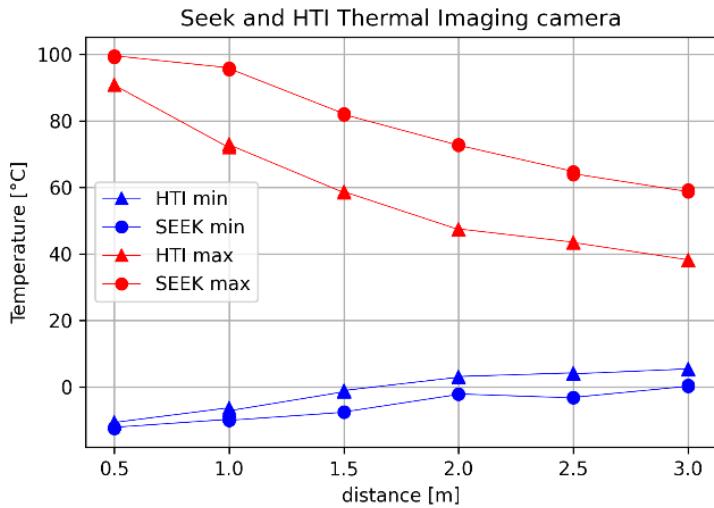


Fig. 31. Comparison between SEEK and HTI temperature measurements at different distance. Each point represents the mean of three replicates.

Then a comparison based on field data collection was done. In this case data collection occurred following the presented apple data collection protocol (Fig. 21), during a hot day (34°C), using as target reference objects a refrigerated container (minimum temperature), the operator's hand (considered as "ambient" temperature -avg-) and one exposed fruit (considered as max temperature). The temperature was then extracted through the implemented system (steps 1-5 in the chapter above) – exploiting a manual labeling of the target object – and compared to HTI point, minimum and maximum collected temperatures. As shown in Fig. 32, minimum and maximum HTI temperatures present low correlation values with all the SEEK collected ones, and this is mainly due to the presence of background-related noise in the analyzed scene. Contrarily HTI point temperature (i.e., the one measured in the center of the scene) results highly correlated with all the SEEK temperatures extracted ($r = 0.93 - 0.98$) (Fig. 32), this because the center of the scene was always and only framing the target objects.

	Tmin_SEEK	Tavg_SEEK	Tmax_SEEK	TpointHTI	TminHTI	TmaxHTI
Tmin_SEEK	1					
Tavg_SEEK	0.994369708	1				
Tmax_SEEK	0.967724864	0.986328732	1			
TpointHTI	0.976161513	0.965340494	0.932539036	1		
TminHTI	0.488047637	0.48311038	0.483338432	0.525161596	1	
TmaxHTI	0.062430662	0.086712329	0.161931994	0.078561197	0.282938777	1

Fig. 32. Correlations between ROS-extracted SEEK and reference HTI temperatures. SEEK – Tmin, Tmax, Tavg are extracted from the object bounding box; HTI – Tpoint, Tmin, Tmax are the temperature readings reported by the thermal camera (N=24)

Further data analysis is currently on going to have a more reliable evaluation, related mostly on fruit temperatures, but preliminary results of the system on grape clusters detection and temperature extraction pointed out an RMSE / mean error ranging from $\pm 3.43 / -0.96$ °C to $\pm 10.36 / -9.79$ °C, for single cluster temperature measured with thermocouples (n=16); the lowest errors occurring when considering maximum fruit temperature extraction and correction. Analyzing the mean fruit temperature extracted for one side of the plant (i.e., the mean of all detected clusters) and comparing it to thermocouples measurements for those plant side (n=6), the RMSE / mean error range from $\pm 3.42 / -1.83$ °C to $\pm 12.20 / -11.97$ °C, with best results again for extraction and correction of maximum temperature. Preliminary results on apples are extracted from fruits measured only through HTI handled thermal camera as reference (N=12), due to the timing of the harvest; in this case results showed an RMSE / mean error ranging from

± 1.38 / -0.95 °C to ± 6.72 / $+6.59$ °C, with best results for extraction and correction of average temperature of the fruit.

The presented preliminary results are quite interesting considering the strong correlation between HTI point temperature and all the ROS-extracted SEEK temperatures; in add, the low errors presented when dealing with maximum-mean fruit temperature, pinpoint an high potentiality of the system for reliable fruit temperature estimation.

Mapping the fruit position in the 3D space

To map fruit position, the depth information coming from the RGB-D sensor was exploited. This camera provides and aligned-to-RGB “depth map” (i.e., a matrix containing distance information for each RGB pixel) from which is possible to extract object related depth information. Before extracting positional information regarding each detected fruit, a filtering step is applied to discard invalid values (i.e., zeros) present due to stereo-induced occlusion, IR light interference or sensor’s noise occurring in the data collection (Fig. 33-a and b).

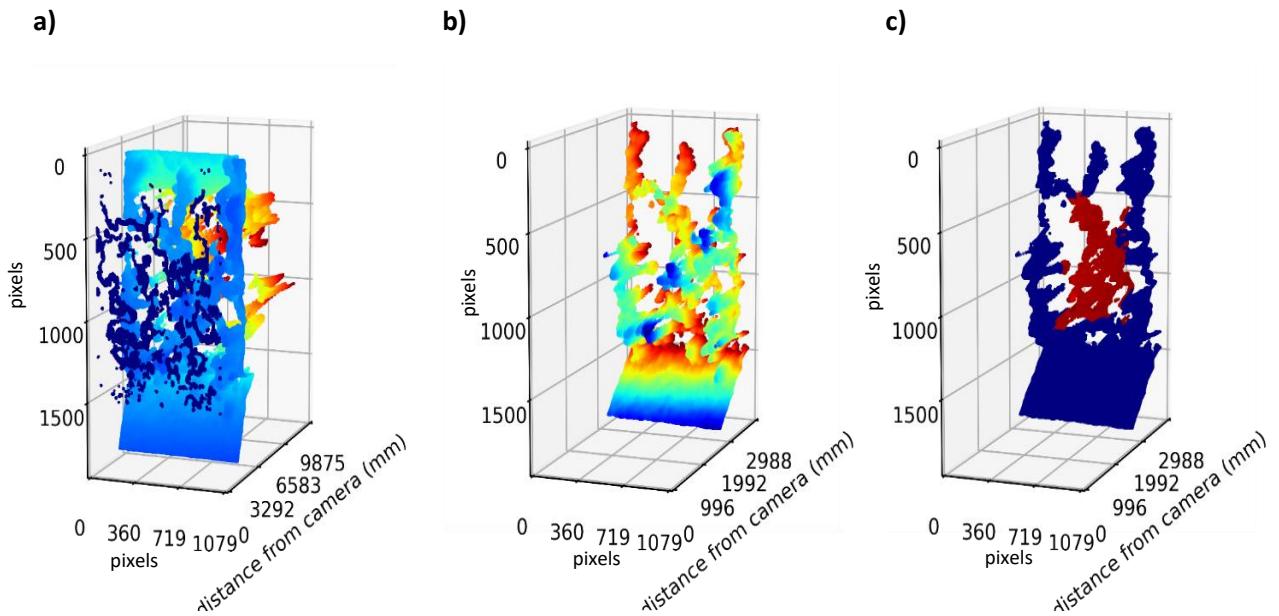


Fig. 33. Representation of an apple tree depth map in a 3D plot; X and Y dimensions represent the pixel coordinates at which depth information (Z dimension – distance from the camera) is stored. A) Unfiltered depth map for null values (i.e., dark blue points). B) Filtered depth map for null values. C) filtered depth map in which are highlighted the points falling in the RGB aligned thermal image (dark red) in respect to the others (dark purple).

After this filtering, as for fruit thermal data extraction, YOLOv5 models are used to detect fruit and trunks (trunk detection model in development), on the RGB image and to clip the aligned depth map for the detected area object (Dbbox).

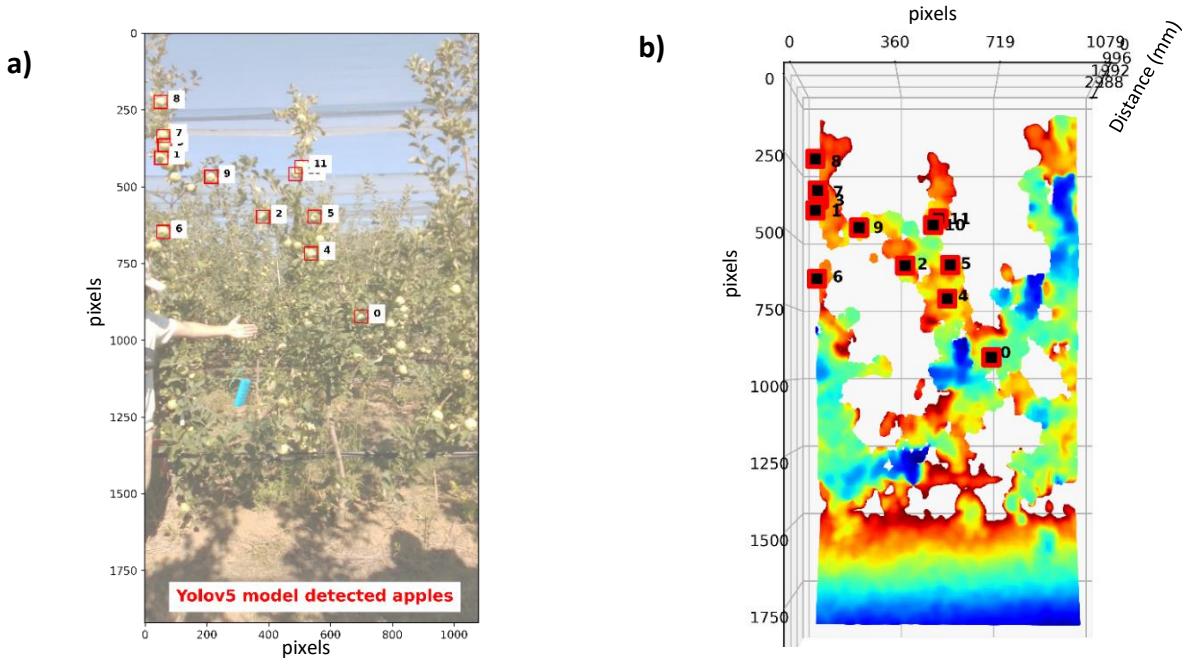


Fig. 34.. fruit detection projection on the 3D depth map representation of a tree. A) RGB image with fruit detection bbox (red rectangle). B) bbox projection on the 3D represented depth map (i.e, Dbbox – red rectangle).

Since the Dbbox is expected to contain a fruit, most of its matrix would contain fruit-related depth information but is possible that background-related depth information is included. This happens, since fruit shapes are not rectangular as the bbox shape (Fig. 28). To account for that, a distance occurrence filtering step is applied to Dbbox: as shown in Fig. 35, after the occurrence filtering application, low frequency (< 10-15%) distance are discarded, while others depth data is maintained.

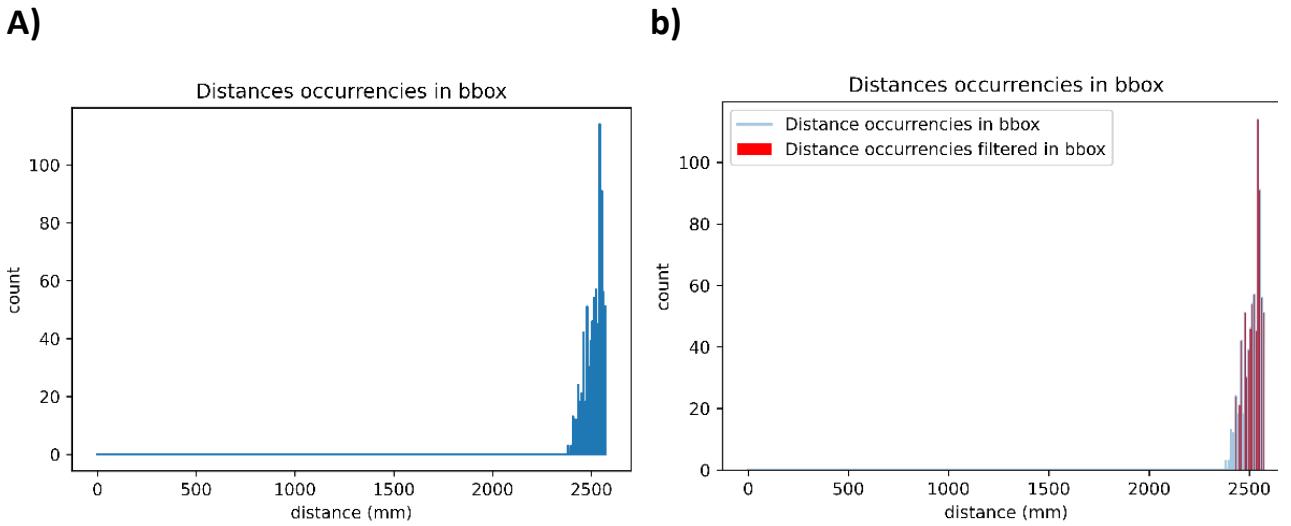


Fig. 35. Pre and post occurrence filtering: a) distance occurrences inside the bounding box (blue); b) filtered distance occurrences (red); the highest bar in the plot is considered as the mean fruit distance

The most represented (i.e., with highest frequency) depth information, is then considered as the “mean fruit distance” (Z coordinate of the fruit) since this should be the one better representing the fruit.

Following, the X and Y coordinates of the fruit center are considered to be the same of the detected fruit bbox center, as shown in Fig. 36 (a,b).

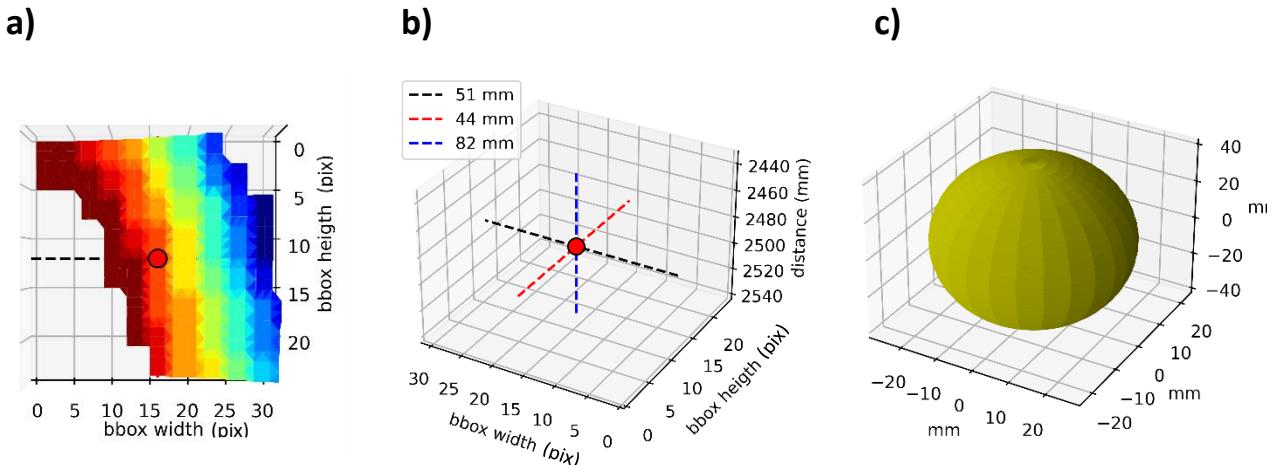


Fig. 36. Steps to obtain 3D coordinates of a detected apple: a) filtered depth map with occurrence filtering: the pixel related to the less frequent distances were removed, it can be seen the remaining distances highlighting the convexity of the fruit; b) location of the estimated fruit center at $X = X$ bbox center, $Y = Y$ bbox center, $Z = \text{mean fruit distance}$; c) 3D reconstruction of a possible apple fitting the cubic bbox obtained by bbox ppro, height and filtered depth information.

Then to obtain the trunk position, the same steps and approaches just exposed was used compute the mean trunk distance (Z), while for X and Y trunk coordinates it was considered the origin of it from the ground, instead of its “center” as done for the fruit. X *bbox center* coordinate of the trunk was considered as X trunk coordinate, while Y trunk coordinate was extracted as Y lowest value in the bbox (i.e., center bbox Y coordinate subtracted of half of the bbox height). Since now the model for trunk detection is still under development, trunk’s bbox were done manually so to mime the object detection algorithm.

Once knowing both the fruit and the trunk coordinates in the same measure unit (X and Y in pixels, Z in millimeters) and from the same coordinate system origin (the RGB-D camera), it is possible to compute the fruit position relative to the trunk position. This was done simply subtracting from each fruit coordinates the correspondent trunk coordinates. Doing that for all the detected fruit in an image, is possible to obtain a fruit map of the whole plant as represented in Fig. 37, Fig. 38.

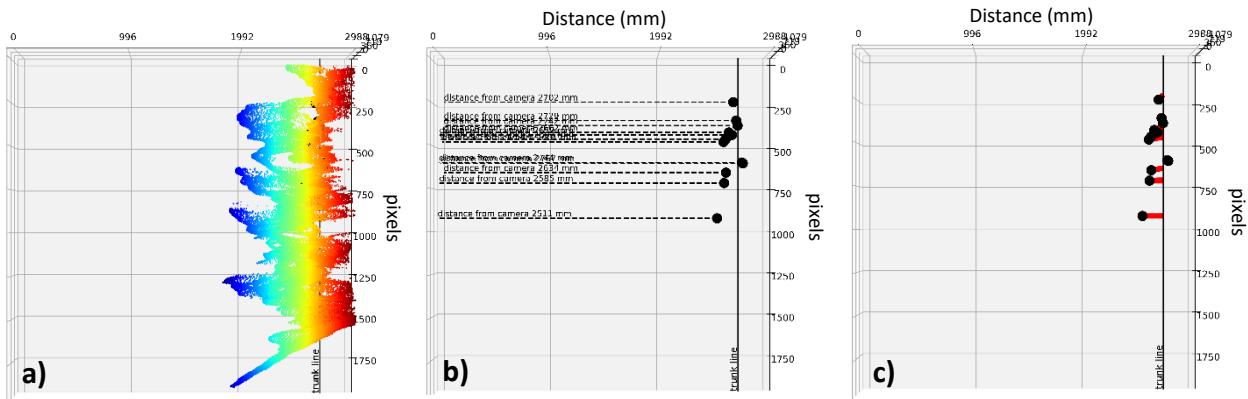


Fig. 37. Step to create a 3D positional fruit map of a tree; in the plots, the main represented dimension are Z (horizontal ax) and Y (vertical ax); vertical black line is the trunk origin position. A) the tree filtered depth map; b) the fruit position extraction with coordinates relative to the RGB-D camera; c) The relative to the trunk fruit position – in red the computed distance/coordinates.

The X and Y pixel coordinates were finally converted in millimeters, to have real world values. This was done exploiting a trigonometric approach accounting for mean object distance and camera’s FoV.

Thanks to the 3D fruit information relative to the single plant trunk, it is possible (in future application) to exploit GPS plant positioning and GIS software to interpolate data coming from plants differently placed in the orchard. A representation of this possibility is shown in Fig. 39.

Evaluation of 3D positional information extraction

The evaluation of the system and process performances in obtaining positional information is currently ongoing. Nevertheless, some very preliminary results pointed out that the system seems to present an RMSE of $\pm 0.1/0.15$ m approx.. in positioning the fruit center in respect to the reference fruit.

We are working to get more precise results as soon as possible.

Conclusion

The presented report described the RGB-D/T-system development and achievements. As anticipated, the intent of this system was to investigate the possibility to use consumer-grade equipment to create a 3D thermal mapping platform of fruit temperature in the orchards. The presented results refer to a first version of the system and can be summarized as follows:

- A thermal-to-RGB alignment with RMSE / mean error of ± 9.17 / $+4.5$ pixels and ± 4.17 / $+0.17$ pixels, on *x-axis* and *y-axis* respectively at 2.60 m distance approx.
- A thermal information extraction process performance presenting a correlation of $r > 0.92$ compared to the thermal reference (after all the exposed conversion and filtering steps) at 2.80 m distance and preliminary results on fruit temperature extraction resulted in a RMSE/mean error ranging in between ± 3.43 : ± 10.36 °C / -0.96 : -9.79 °C, for grape (at 2.30 m distance), and ± 1.38 : ± 6.72 °C / -0.95 : $+6.59$ °C, for apples (at 2.80 m distance). A preliminary fruit 3D positional error of 0.10/0.15 m approx. at 2.80 m distance.
- Object detection model performances are still under evaluation for their 2nd round of training , but resulted satisfactory to reach the presented results.

Further testing and development will be carried out to improve the system performances, but as exposed the presented results are highly probably confirming the possible utilization of the developed system for the purpose of the study. The figures below show a representation of the results of the whole developed process.

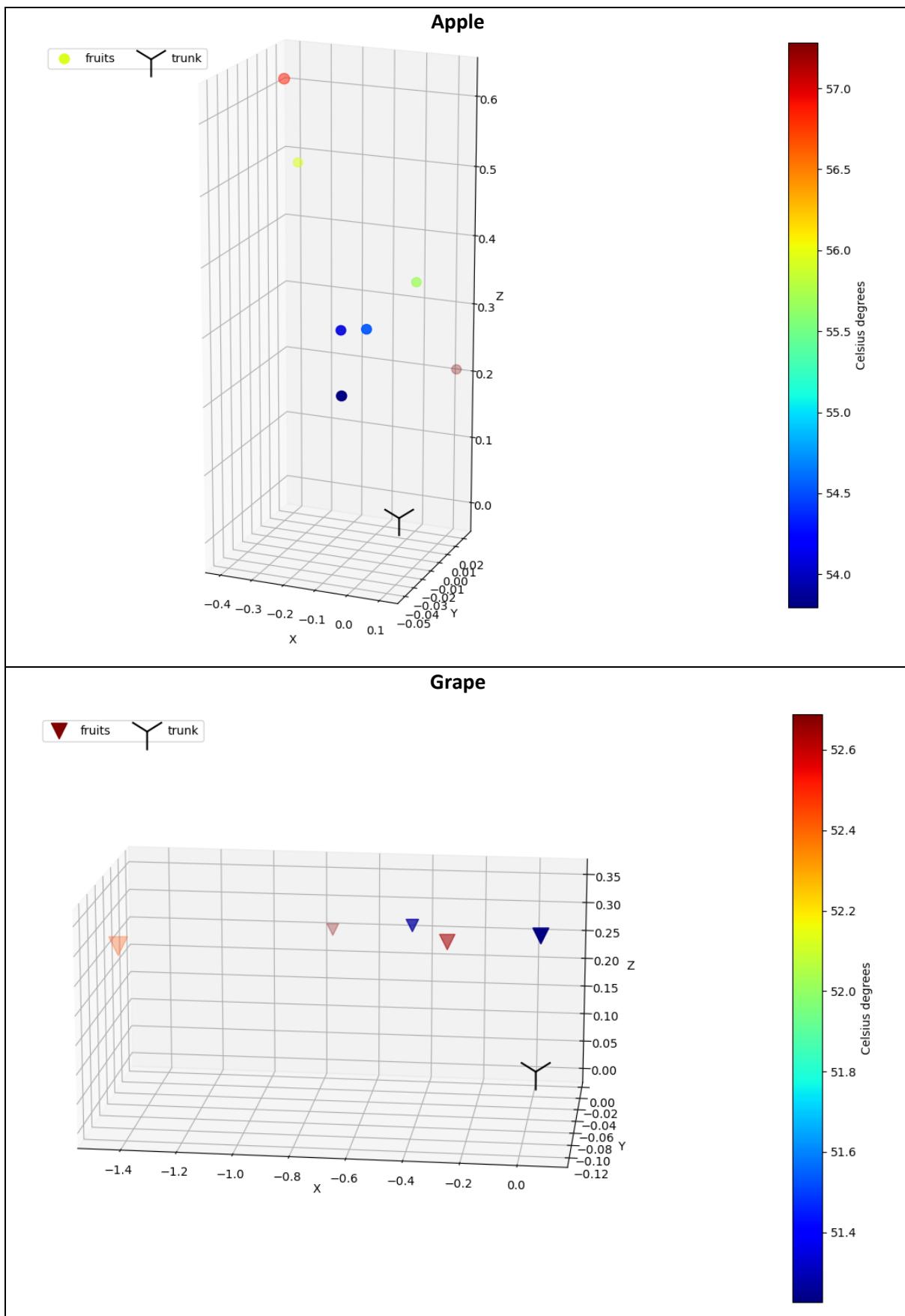


Fig. 38. Fruit temperature 3D representation with relative position in respect of the trunk origin. Coordinates are relative to the trunk.

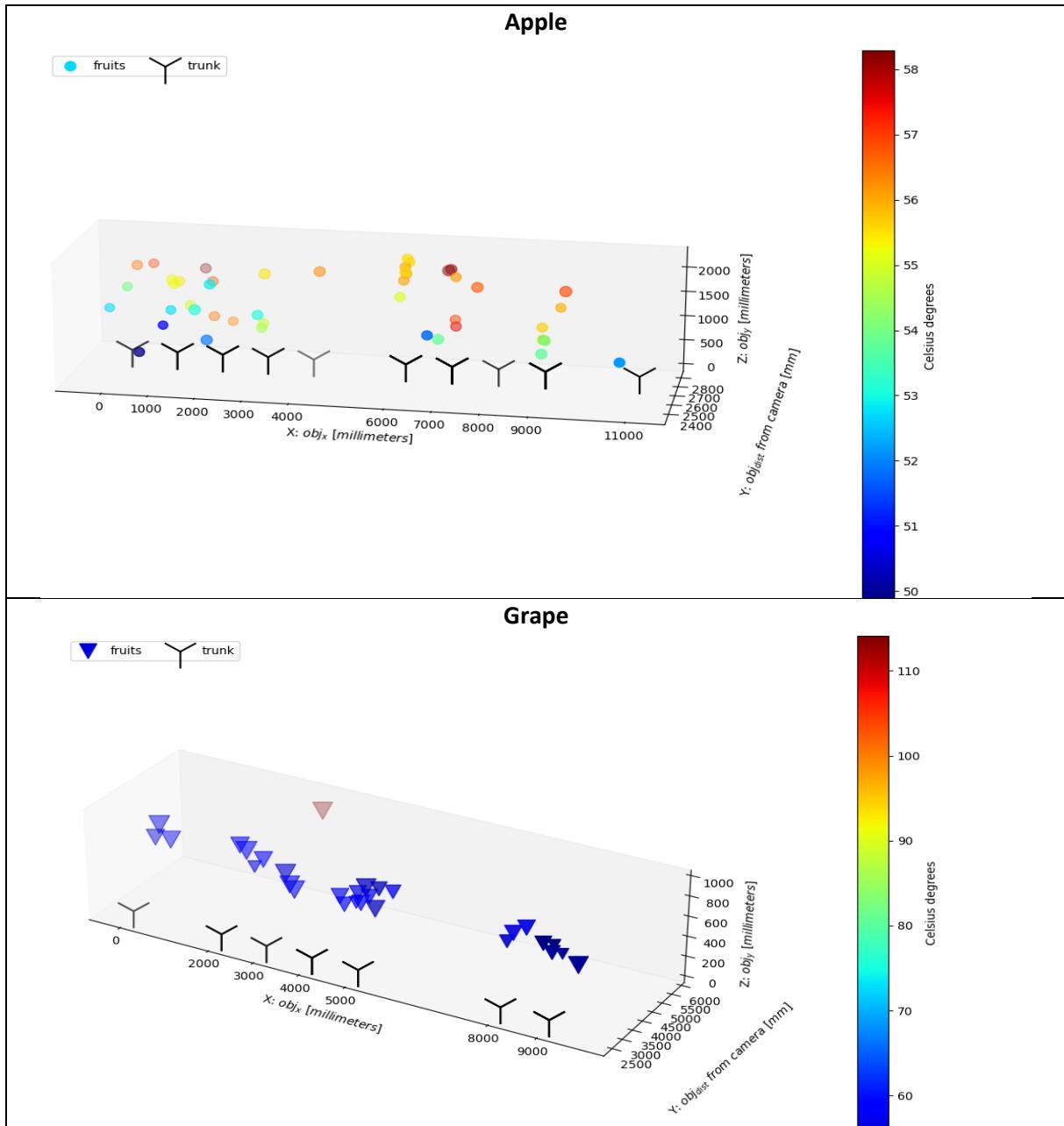


Fig. 39. Orchard fruit temperature 3D representation with fruit position relative to a defined orchard origin. Coordinates are in millimeters and relative to the set orchard origin.

References

- [1] N. Tsoulias, S. Jörissen, and A. Nüchter, 'An approach for monitoring temperature on fruit surface by means of thermal point cloud', *MethodsX*, vol. 9, Jan. 2022, doi: 10.1016/j.mex.2022.101712.
- [2] D. Mengoli, G. Bortolotti, M. Piani, and L. Manfrini, 'On-line real-time fruit size estimation using a depth-camera sensor', in 2022 IEEE Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Nov. 2022, pp. 86–90. Doi: 10.1109/MetroAgriFor55389.2022.9964960.
- [3] G. Bortolotti, D. Mengoli, M. Piani, L. C. Grappadelli, and L. Manfrini, 'A computer vision system for in-field quality evaluation: preliminary results on peach fruit', in 2022 IEEE Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Nov. 2022, pp. 180–185. Doi: 10.1109/MetroAgriFor55389.2022.9965022.

Appendix-1 – Python Code “HOWTO”

After the data field collection with the RGB-D/T system, synchronized extracted images need to be placed respectively in the RGB, thermal (TERM) and depth image directories (“_dir”). Equally, the resulting “.txt” labels files obtained by applying the object detection models on the RGB extracted images need to be placed in a dedicated directory (“Yolo_fruit_labels_dir” and “Yolo_trunk_labels_dir”).

If not already done, exploiting the images obtained through the utilization of the thermal alignment panels, alignment coordinates and scaling factors (both saved in a dedicated “.txt” file) need to be extracted through the “Get_alignment_factors.py” script before proceeding to fruit temperature and position extraction.

The developed python program takes from an input file (INPUTS_[specie].py) the directory of RGB, thermal and depth images, as well the camera location information in the orchard (CAM_loc – including information of distance from trees, height, geographic orientation) and the fruit/trunk detection labels directories.

Data are then used to align the images (exploiting the previously obtained Alignment_coords.txt and Scale_factors.txt) and to calculate the detected fruit temperature (Thermal_sheet.py \otimes Thermal_correction.py) and positioning them in the 3D – space (XYZ_positioning_main.py).

Results (Output_dataset.csv) and some graphical representation will be saved (Visualize_orchard.py).

The code is available at https://github.com/ECOPOM/SHEET_RGBD-T_system_v_2022

SHEET_project_tree_structure

```
 ALIGNMENT.py
 Get_alignment_factors.py
 INPUTS_apple.py
 INPUTS_grape.py
 RUN_file.py
 SHEET_Temp_Position_extractor_from_Yolo.py
 Thermal_correction.py
 Thermal_sheet.py
 Visualise_orchard.py
 XYZ_functions.py
 XYZ_positioning_main.py
```

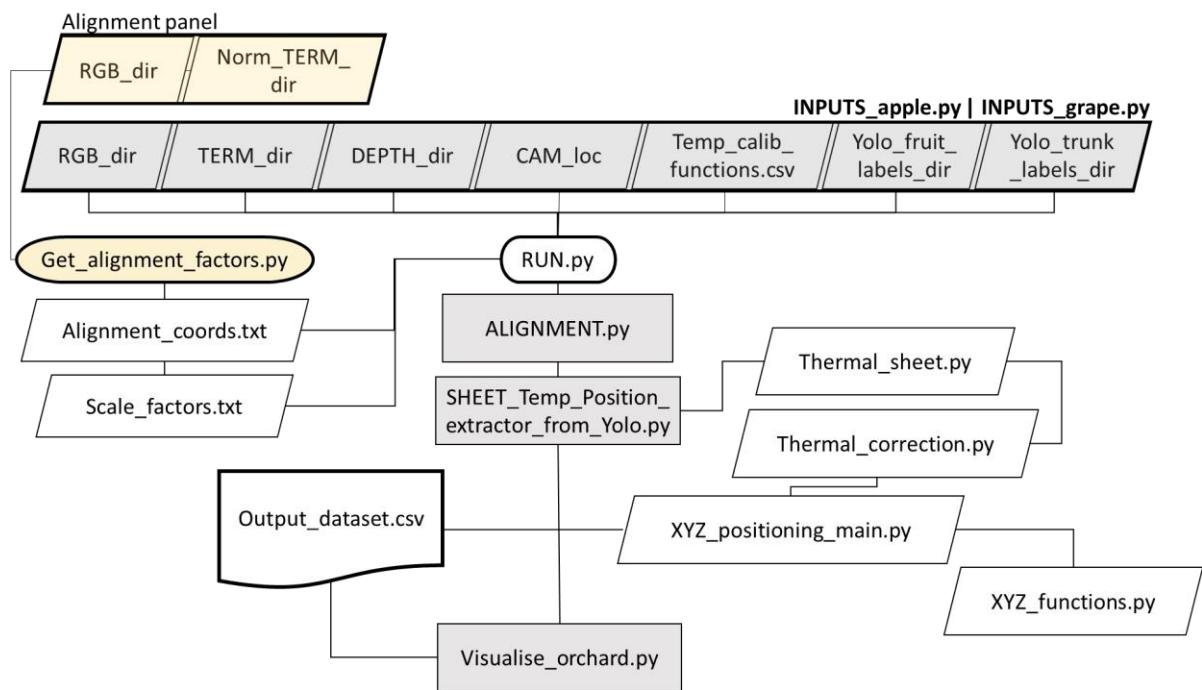


Fig. 40. Python program workflow

Deliverable No D2.4

Deliverable hypothesis

Deliverable No	Hypothesis
D2.4	<ul style="list-style-type: none">• Apple fruit can be segmented from LiDAR and RGB-D 3D point cloud of trees in the orchard.• Cherry fruit can be segmented from 3D point cloud of trees in the orchard.• Grape can be segmented from 3D point cloud of trees in the orchard.

Deliverable Description

Deliverable No	Description
D2.4	Fruit position and temperature data (preliminary) are open access and available as input for further modelling in the three fruits (apple, grape, sweet cherry).

In relation to D2.4, different fruit datasets were collected from UNIBO and ATB partners.

UNIBO – Apple temperature and position datasets

Treatments

In relation to thermal and positional information of apple fruit (*Malus domestica*), 2 main plus 1 preliminary datasets (main: Dataset 1 or “Tcouple” and Dataset 2 or “ThCAM”; preliminary: Dataset 3 or “RGB-D/T-system”) were produced using different methodologies. For dataset 1 and 2, fruit temperature measurements were taken in 4 different training system*shading treatments (Fig.41). The treatments were:

A: Spindle training system (3.3 m x 1 m planting distances) * Black anti-hail net coverage (15-20% shading). Considered as “Standard Sunburn Susceptibility Situation” (N=4 trees).

E: Spindle training system (3.3 m x 1 m planting distances) *White anti -hail, -insect, -rain exclusion netting system (40-50% shading). Considered as “Low Sunburn Susceptibility Situation” (N=4 trees).

P: Planar Cordon training system (2 m x 3 m planting distances) *Grey anti-hail net coverage (15-20% shading). Considered as “Mid-High Sunburn Susceptibility Situation” (N=3 trees).

RST: Spindle training system (3.3 m x 1 m planting distances) * Open field (no netting / shading applied). “High Sunburn Susceptibility Situation” (N=4 trees).



A



E



P



RST

Fig. 41. Treatments' conditions: **A** – “Standard Sunburn Susceptibility Situation”; **E** – “Low Sunburn Susceptibility Situation”; **P** – “Mid-High Sunburn Susceptibility Situation”; **RST** – “High Sunburn Susceptibility Situation”

The purpose of comparing these treatments was to investigate different environmental conditions in relation to sunburn damage susceptibility, collecting a wide range of data that can support further modelling and investigations of temperature, fruit position and sunburn damage.

For the preliminary Dataset 3, data collection occurred in a different orchard, not in trial, with similar environmental condition as the one of treatment A.

Fruit localization

The GPS coordinates (Latitude – ‘Lat.’, longitude - ‘Long.’, altitude - ‘Altitude’) of trees (N=15) carrying fruit under analysis were collected and stored, on the ATBCloud, in the file named “GPS_Position_Apple&Tree_SHEET_2022_OK_EPSG4326.xlsx” (Fig. 42).

TRT (identifier)	Plant (identifier)	Lat. (decimal °)	Long. (decimal °)	Altitude (m)	GPS_log
1 A	1	44.549341	11.4163637	27.4128609	3
2 A	2	44.5493493	11.4163684	27.3893509	2
3 A	3	44.5493572	11.4163747	27.3370552	1
4 A	4	44.5493658	11.4163791	27.3224678	0
5 E	1	44.5492782	11.416566	27.43153	4
6 E	2	44.549286	11.4165722	27.4390678	5
7 E	3	44.5492946	11.4165769	27.3883114	6
8 E	4	44.5493025	11.416583	27.3965931	7
9 P	1	44.5475346	11.4140268	27.617794	11
10 P	2	44.5475583	11.4140436	27.5253334	10
11 P	3	44.5475837	11.414065	27.4316444	8
12 RST	4	44.5495055	11.416156	26.7332668	12
13 RST	3	44.5495082	11.4161608	26.7332668	13
14 RST	2	44.5496618	11.41628	26.9152107	14
15 RST	1	44.5496672	11.416286	26.8998127	15

Fig. 42. 'Apple_TREE_GPS position' sheet in "GPS_Position_Apple&Tree_SHEET_2022_OK_EPSG4326.xlsx" file

In the same file, for all the monitored fruit in datasets Tcouple (N=40) and ThCAM (N=108), their position was collected as X, Y, Z coordinates, in centimeter, using a standard measuring tape. X, Y, Z dimensions were considered respectively as the tree-row plane, the vertical plane perpendicular to the ground and the intra-row tree plane perpendicular to X (Fig. 43). Positive and negative values of the coordinates were representing the fruit position in relation to the trunk (coordinate system's origin), with Y = 0 at ground level, +X or -X values for fruit positioned toward North or South respectively, and +Z or -Z values for fruit positioned toward East or West respectively, in respect to the trunk origin.



Fig. 43. X, Y, Z coordinate system representation; the origin is represented by the trunk, with Y=0 at ground level.

Fig. 44 below shows the file structure for both thermocouple and tagged fruit (i.e., fruit in ThCAM-dataset) position information. In add to 'X', 'Y', 'Z' coordinates and trees' GPS coordinates ('Lat.', 'Long.', 'Altitude'), and identifier ('Plant'), other information related to treatments ('Trattamento' and 'TRT') and internal identifier for tree position and block ('Block') are present. 'ThCouple_Node' and 'Th.couple_N' columns show if the fruit was equipped with a thermocouple (and if so, its number and the mounting node identifier of the WSN), while 'FruitTAG' one, represents the unique identifier of each fruit. 'Side', 'Canopy_Height_zone' and 'Canopy_Width_zone' columns represent the East or West side and the zone of the canopy in which the fruit was positioned.

Thermocouple position														
Trattamento (Identifier)	Block (Identifier)	TRT (Identifier)	ThCouple_Node (Identifier)	Th.couple_N (Identifier)	FruitTAG (Identifier)	Plant (Identifier)	Lat. (decimal °)	Long. (decimal °)	Altitude (m)	X (cm)	Y (cm)	Z (cm)	Side (Identifier)	
Planar	Fila6	P	000D	1		1	44.547535	11.414027	27.61779404	95	122	0	e	
Planar	Fila6	P	000D	2	101	1	44.547535	11.414027	27.61779404	-101	144.5	-4.5	w	
Planar	Fila6	P	000D	3		1	44.547535	11.414027	27.61779404	137.5	158	-6.5	w	
...	
...	
...	
Rootstock	2	RST	000E	6		4	44.549506	11.416156	26.73326683	7	80	-4.5	w	
Rootstock	2	RST	000E	7	67	3	44.549508	11.416161	26.73326683	47.5	106.5	-50	w	
Rootstock	2	RST	000E	8		3	44.549508	11.416161	26.73326683	14.5	84	-27	w	

Tagged fruit position															
Trattamento (Identifier)	Block (Identifier)	TRT (Identifier)	Lat. (decimal °)	Long. (decimal °)	Altitude (m)	Plant (Identifier)	Side (Identifier)	Canopy_Height_zone (Identifier)	Canopy_Width_zone (Identifier)	X (cm)	Y (cm)	Z (cm)	ThCouple_Node (Identifier)	Th.couple_N (Identifier)	FruitTAG (Identifier)
Rootstock	3	RST	44.5496672	11.416286	26.8998127	1	e	low	out	-30	55	36			1
Rootstock	3	RST	44.5496672	11.416286	26.8998127	1	e	low	in	-8.5	70	2.5			2
Rootstock	3	RST	44.5496672	11.416286	26.8998127	1	e	mid	out	19	128	26.5			3
...
...
...
Planar	Fila 6	P	44.5475837	11.414065	27.4316444	3	w	low	out	68	73.5	0			106
Planar	Fila 6	P	44.5475837	11.414065	27.4316444	3	w	mid	out	-33	134	0			107
Planar	Fila 6	P	44.5475837	11.414065	27.4316444	3	w	high	out	9	180	-20			108

Fig. 44. Example of the structure of the 'ThCouple_Fruit Position' and 'Tagged_Fruit_Position' sheets present in the "GPS_Position_Apple&Tree_SHEET_2022_OK_EPSG4326.xlsx" file

For preliminary Dataset 3 no fruit positions were collected manually as reference.

Sunburn symptoms evaluation

For fruits of datasets Tcouple (N=40) and ThCAM (N=108) only, the occurrence of sunburn was evaluated 1-2 times per week (from 64 days after full bloom -DAFB-, occurred on 10th of April, till the harvest occurred on 114 and 120 DAFB – i.e., 02/08 and 08/08), so to individuate when first sunburn symptoms appeared ('Sunburn Damage' in Fig. 45). During the season (at 95, 102 and 109 DAFB), also color and chlorophyll degradation information were collected using a Minolta Colorimeter (CR400, Konica-Minolta, Japan) and a DA-Meter (DA-Meter, Sinteleia srl, Bologna – Italy) , on a sample of the fruit in trial (N = 24 to 40 in base of the treatment). At harvest, a sample of fruit (N = 33 to 51 in base of the treatment) was evaluated again for color (i.e., 'L*(C)', 'a*(C)', 'b*(C)', 'C*(C)', 'h(C)' color coordinates in Fig. 45) and chlorophyl degradation ('DA-meter', in Fig. 45), as well as for a more in-detail sunburn damage presence ('Sunburn level', in Fig. 45); this to give valuable information for further investigation and modelling of apple sunburn induction and relation with fruit and environmental temperature dynamics. Data regarding sunburn symptoms evaluated both during the season and the harvest are stored in the "Sunburn_data_SB+Minolta+Dameter_2022.xlsx" file, on the ATBCloud (Fig. 45).

Date (dd/mm/yyyy)	Sunburn Damage (Identifier)	Sunburn level (damage scale)	L*(C)	a*(C)	b*(C)	C*(C)	h(C)	DA-meter
14/07/2022			70.23	1.68	38.51	38.5466276	85.2685054	1.377
14/07/2022			72.15	-7.125	41.5	42.1071921	99.7458657	1.545
...
...
21/07/2022	s		63.99	14.355	32.31	35.53	66.61	1.115
21/07/2022	s		58.16	15.925	33.64	37.46	64.625	1.373
21/07/2022	s		56.64	25.685	25.48	36.235	44.895	1.234
...
...
02/08/2022	s	sb1-3	64.24	15.11	38.59	41.475	68.585	
02/08/2022	s	sb1-3	59.9	20.13	33.04	38.7	58.655	0.8006
08/08/2022			59.81	21.33	35.23	45.285	58.4	0.3397
08/08/2022	s	sb1-3	51.31	31.77	27.72	42.155	41.105	0.2885

Fig. 45. "Sunburn_data_SB+Minolta+Dameter_2022.xlsx" file structure: in add to all the fruit localization information, the file contains the columns shown in the picture and described in the text.

In Fig. 45 above, are shown the sunburn related information, present in the file in add to the positional fruit information already described in Fig. 44. Columns were described in the text above if not for 'Date' which represent the date of data collection. Detailed information can be found in the 'README-Legend' sheet of the described file.

Fruit temperature and weather data

For all the presented datasets, seasonal local weather data were available and shared ("Weather_data_IN&OUT_Orchards.xlsx" file on ATBCloud).

Fruit temperature data were collected following different methodologies as presented below in each of the dataset description.

Dataset 1 (Tcouple) – Continuous thermal measurements

From 64 DAFB (14/06) to fruit harvest 114 DAFB (02/08), Gala apple fruit were continuously monitored (data resolution 15 min, data presented as hour average) for their temperature using calibrated thermocouples (Tcouple) (model: 'Type K' Tcouple, WiNet srl, Cesena – Italy) connected to a wireless sensor network (WSN) (WiNet srl, Cesena – Italy).

For treatments A, E, P, four highly exposed fruit on each side (east and west) were selected, at medium height, for monitoring their temperature, on 4 different plants (N = 8 per treatment, 2 per tree). For RST treatment the fruit number was doubled (N = 16 on 4 per trees).

Tcouple were fixed to the fruit using medical tape on the back / less exposed fruit side to keep cable in position, while the Tcouple itself was maintained in position thanks to the high cable plasticity.

Tcouple was always touching the fruit, without damaging it, as shown in Fig. 46. Tcouple was checked 1 – 2 per week for their correct positioning. This dataset was uploaded on the ATBCloud as "Th.couple_data_2022_timeseries.xlsx" file.



Fig. 46. Detail of thermocouple mounted on apple fruit

1	TRT	RST	RST	RST	RST	RST	RST	RST	RST	
2	ThCouple_Node	000B	000B	000B	000B	000B	000B	000B	000B	
3	Side	W	W	W	W	E	E	E	E	
4	Canopy_Width_zone	int	ext	ext	int	int	ext	int	ext	
5	Th.couple_N									
6	Date	time	1	2	3	4	5	6	7	8
7	14/06/2022	00:00:00	15.95	18.27	16.20	17.25	17.03	17.63	17.20	17.40
8	14/06/2022	01:00:00	15.38	17.28	15.58	16.43	16.55	17.05	16.53	16.47
9	14/06/2022	02:00:00	15.25	16.88	15.55	16.17	16.30	16.80	16.37	16.22
...	
29	14/06/2022	22:00:00	18.00	19.90	18.40	19.10	19.15	19.75	19.33	19.15
30	14/06/2022	23:00:00	16.53	18.60	16.80	17.68	17.23	18.15	17.67	17.82
31	15/06/2022	00:00:00	14.77	17.00	15.05	15.98	16.23	16.72	16.05	16.12
32	15/06/2022	01:00:00	14.25	16.40	14.63	15.55	15.93	16.17	15.68	15.63
33	15/06/2022	02:00:00	14.00	16.02	14.27	15.20	15.20	15.73	15.28	15.23
...	
1161	01/08/2022	23:00:00	21.15	22.98	20.93	21.98	22.38	22.83	22.40	22.00

Fig.47 Tcouple dataset structure

The dataset presents, per each treatment, the structure shown in Fig. 47 above: From row 1 to 5, are reported information related to treatment ('*TRT*'), WSN node utilized ('*ThCouple_Node*'), mounting '*Side*' (row 3 – East or West) and '*Canopy_Width_zone*' (row 4 – external or internal) of the Tcouple on the tree. Row 6 report the "*Date*" and "*time*" columns of the timeseries, while columns "1" to "8" represent the hour average of fruit temperature, in Celsius degrees, recorded each of the 1-to-8 mounted Tcouple. Fruit positions were not included in this dataset directly, as well as sunburn damage, but they are stored in the different files described above, which can be linked by the unique *ThCouple_Node** '*Th.couple_N*' values .

Dataset 2 (ThCAM) – Discrete thermal measurements.

This dataset was created manually measuring fruit temperature of 27 tagged fruit in each of the A, E, P and RST treatments. In each treatment, the 27 fruits were chosen, on three plants (9 per plant), at 3 heights (low -mid- high) and 3 canopy positions (west side -inner part -east side). The total number of monitored fruit was 108.

Fruit temperature data were collected 5 times during the 2022 season (93, 101, 112, 119 and 131 DAFB). Per each time, temperature collection occurred three times per day (morning -midday -afternoon) in a time window of 1.5h approx. to collect all 108 tagged fruit data.

Temperature measurements were collected utilizing a semi-professional grade thermal camera (HTI-HT-A9, Xintai Instrument Co., Ltd., China; <https://hti-instrument.com/products/ht-a9-thermal-imager>). Temperatures were carefully collected trying to always frame the fruit in the scene center, at 30-50 cm distance max, and placing the thermal pointer to collect temperature on the most exposed area of the fruit, considering its position (Fig. 48).

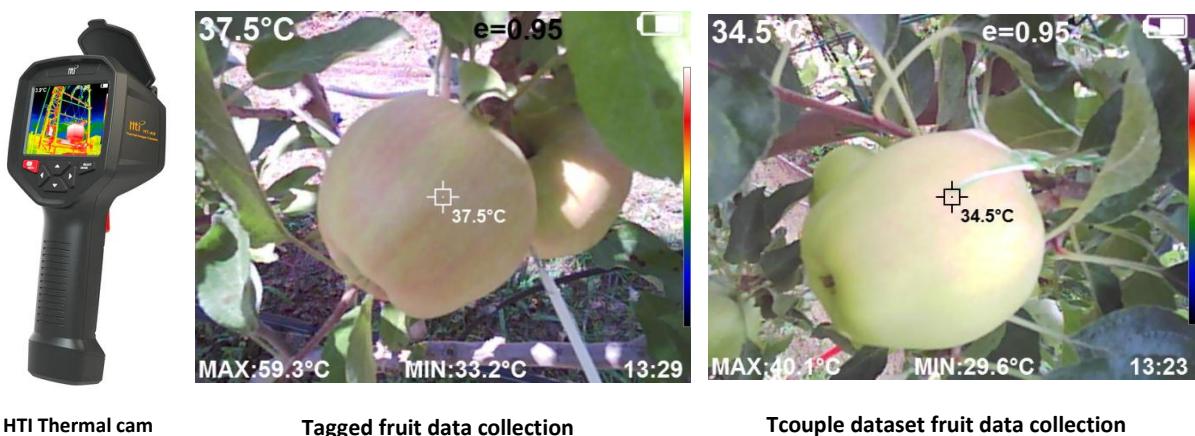


Fig. 48. From left: Detail of the thermal camera utilized; Detail of temperature collection of a tagged fruit without Tcouple; Detail of temperature collection of fruit equipped with Tcouple.

During these measurements days, fruit of the Tcouple dataset were also measured with the HTI thermal camera to investigate possible correlations or errors between the two utilized sensors (Fig. 48). Considering Tcouple as the reference, the comparison of these sensors resulted with a RMSE (root mean square error) = ± 5.04 °C, and a mean error = + 4.14 °C. A regression analysis performed on the collected data showed a $R^2 = 0.775$. The obtained regression model was used to correct the thermal camera reading and resulted in halving RMSE (= ± 2.17 °C) and nulling the mean error (= +0 °C). Further investigations will be done searching for correlation between temperature estimation errors and air temperature which could have altered ThCAM readings.

This dataset was uploaded on the ATBCloud as “SHEET_Th.CAMHTI_2022.xlsx” file.

0	Date (dd/mm/yyyy)	start time (hh:mm:ss)	end time (hh:mm:ss)	timeTRT (identifier)	Sunburn Damage (identifier)	ThermalCAM (°C)	ThCAM -Corrected (°C)
1	22/06/2022	09:45:00	11:35:00	morning		27.00	25.97
2	22/06/2022	09:45:00	11:35:00	morning		26.30	25.49
3	
...
155	19/07/2022	13:05:00	14:05:00	midday		45.70	38.68
156	19/07/2022	13:05:00	14:05:00	midday		39.50	34.47
...
...
1404	01/08/2022	15:45:00	17:00:00	afternoon		43.80	37.39
1405	01/08/2022	15:45:00	17:00:00	afternoon		42.60	36.57

Fig.49 ThCAM dataset structure

in Fig. 49 are shown the adding information to all the fruit localization data described above (Fig. 44), for the ThCAM dataset. ‘Date’, ‘start time’ and ‘end time’ show the date and time frame in which fruit temperatures were collected; ‘timeTRT’ represent the timing treatment as morning, midday, afternoon time. ‘Sunburn Damage’ shows when a sunburn symptom was identified in field for the first time (with no classification level). ‘ThermalCAM’ and ‘ThCAM -Corrected’ are respectively fruit temperature values collected with HTI thermal camera and then corrected with the extracted regression aforementioned (i.e., ThCAM vs TCouple). This information can be found in the *ThermalCAM_measurements* sheet of the file.

In the same dataset file, a second sheet is included (called *ThCAM_vs_Thcouple Calib*), where fruit temperature of only those fruit equipped with thermal couple was collected to obtain the just mentioned regression model. In this sheet, in add to the all the columns just presented, ‘*T_ThCouple*’ shows the temperature recorded by Tcouple for the same fruit and timing of HTI thermal camera measurements; metrics computed between Tcouple and ThCAM measurements are then presented as absolute and percentual errors (respectively ‘*T_Error (ThCAM-ThC)*’ and ‘*T_Error%(ThCAM-ThC)*’), and then the related average errors (‘*meanERR*’ and ‘*meanERR%*’) and RMSEs (‘*RMSE*’, ‘*RMSE%*’). In add absolute error and related average error and RMSE were computed after ThCAM temperature regression correction (respectively ‘*T_ErrorADJ(ThCAM-Corrected-ThC)*’, ‘*meanERR_ADJ*’, ‘*RMSE_ADJ*’).

More in-detail description can be found in the dataset file directly, in the sheet named “*README-Legend*”.

Dataset 3 (RGB-D/T-system) – Fruit thermal scanning dataset – PRELIMINARY

This preliminary dataset was obtained through the utilization of the RGB-D/T-system developed and previously presented in chapter “Deliverable No D2.2 – Continuation” (pag. 25), in which also the utilized data collection protocol and process analysis were explained.

To summarize, RGB-D/T-system utilized consist in a sensor fusion platform, based on a depth and a thermal camera, from which were extracted single fruit thermal and positional information. The RGB-D/T system performances resulted in:

- A thermal-to-RGB alignment RMSE / mean error of ± 9.17 / $+4.5$ pixels and ± 4.17 / $+0.17$ pixels, on *x-axis* and *y-axis* respectively;
- A thermal information extraction process presenting a correlation of $r > 0.92$ compared to the thermal reference at 2.80 m distance and preliminary results on fruit temperature extraction resulted in a RMSE/mean error ranging in between ± 3.43 : ± 10.36 °C / -0.96 : -9.79 °C, for grape (at 2.30 m distance), and ± 1.38 : ± 6.72 °C / -0.95 : $+6.59$ °C, for apples (at 2.80 m distance);
- A preliminary fruit 3D positional error of 0.10/0.15 m approx. at 2.80 m distance.

The presented dataset was obtained for apple fruit not related to the above presented treatments and / or datasets but placed in a similar environmental condition as the treatment A. Since the processing pipeline to align and merge data extracted from both “h1” and “h2” recordings (see “Field data collection with the RGB-D/T system” subchapter, pag. 30) is still under development, data in this dataset are related only to “h1” recordings, where the trunk of the tree was visible in each picture.

The dataset was uploaded on the ATBCloud as “RGBD-T-System_APPLE_Pos&Temp_Dataset.xlsx” file and it includes only information of those fruits correctly detected and that passed all the process filtering steps explained in “Deliverable No D2.2 – Continuation”. In total, 10 out of 12 tree images were correctly analyzed, obtaining thermal and positional information for a total of 46 fruit. The two missing images were not analyzed due to the non-falling in the thermal camera field of view of the detected fruits and/or no fruit detection at all. A graphical representation of the obtained results is present in Fig. 50, reported below.

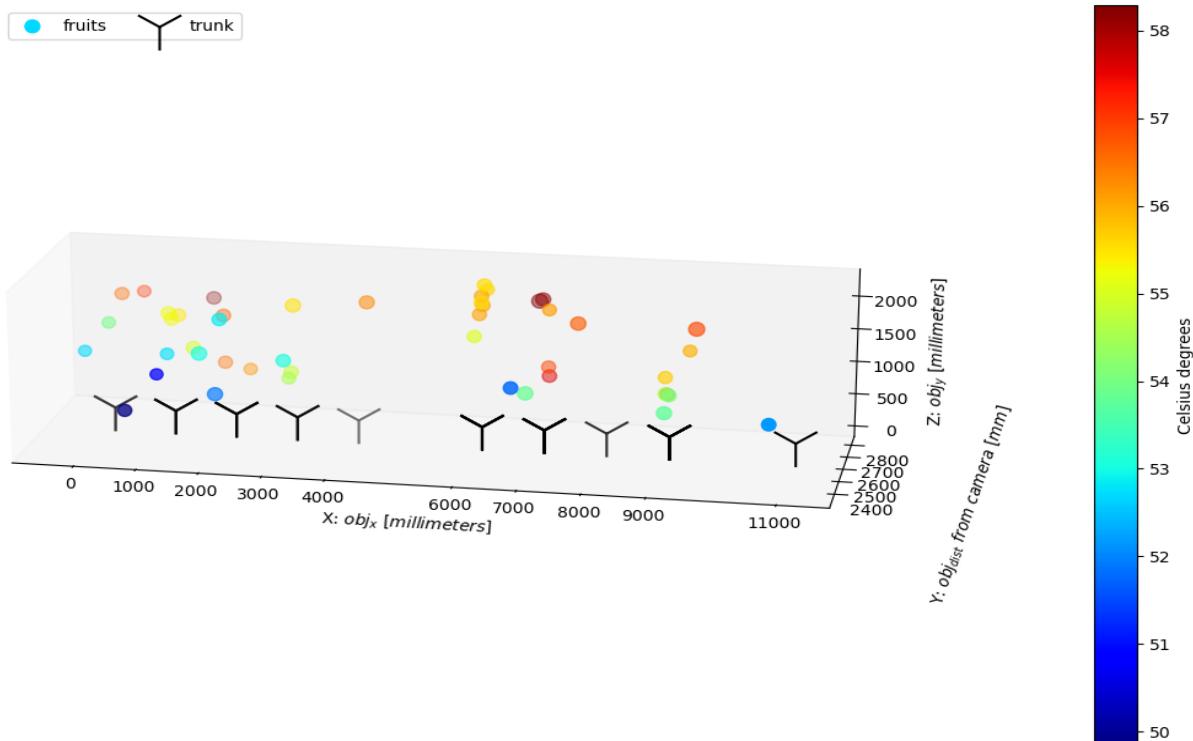


Fig. 50. Copy of Fig. 39 -Apple of the previous chapter, representing apple fruit temperature 3D representation at “orchard level”, with fruit position relative to a defined orchard origin, and temperature represented by color-scale.

As already explained reference data (i.e., actual fruit temperature and position) to test system performances on this datasets were missing due to the late development of the RGB-D/T system; so no performances evaluation for data collected in this dataset can be presented.

Due to the dimension of the dataset, no picture can be presented here, but the RGB/T-System dataset is structured in seven different sections (*sec0* to *sec6*) which present, per each detected fruit, what follows:

- *Sec0* presents the ‘name’ of the analyzed image and its internal identifier (‘*ID_tree*’) as well as the ‘*label*’ class number of the object detected by the YOLO algorithm used for fruit detection;
- *sec1* reports minimum (min), mean and maximum (max) temperature directly extracted from the thermal camera after the “D2.2 – Continuation” process (respectively, ‘*Tmin_SEEK*’, ‘*Tmean_SEEK*’, ‘*Tmax_SEEK*’);
- *sec2* reports six different min, mean and max temperature, those result from the distance correction of *sec1* data according to each of the two correction equations utilized (‘*corr_avg*’ and ‘*corr_max*’);

- *sec3* contains 'X', 'Y', 'Z', *coordinates in millimeters* ('_mm') relative to an arbitrary orchard origin (of 0,0,0 mm coordinates), both for detected fruit and tree trunk ('_trunk_'), to work at orchard level scale, plus the *original distance measured by the depth camera* ('Z_{cam}') both for fruit and trunk;
- *sec4* presents X, Y, Z *coordinates relative to the trunk origin* (with absolute values) for the analyzed fruit, as well as the trunk origin (i.e., 0, 0, 0);
- *sec5* presents the X, Y, Z *relative ranges* used to convert the fruit localization from relative to mm, and viceversa, allowing also the change between single tree and orchard scale level;
- *sec6* shows additional information as tree side of data collection ('Tree_wall' – east or west), the cardinal orientation toward the X-axis ('Card_dir') and a preliminary estimated fruit size ('Estim_fruit_diam_mm') useful for further implementation of the system.

A more in-detail description is present inside the dataset file in the '*readme*' and '*REFERENCE crs*' sheets.

Considering the preliminary testing of the platform (performances reported above), due to the current possible high error of the system (up to ± 10.36 °C and ± 0.15 m), we suggest avoiding using data from this dataset to train model for investigating sunburn dynamics. However, this preliminary dataset is presented, and shared, so to have feedbacks for possible improvements and to make other partners able to evaluate it.

UNIBO – Grape temperature and position datasets

Treatments

The temperature data and the position of the monitored grape berries were reported in two datasets produced using different methodologies. Those data were taken in a trial conducted in 2022 in a vineyard of the black-berry variety Sangiovese (*Vitis vinifera* L), where the treatments were laid out in a strip-plot design and the main factors were cluster exposure and irrigation. Cluster exposure treatments were: leaf removal of the basal leaves of each shoot at the beginning of veraison (58 days after full bloom – DAFB-) and no leaf removal. Irrigation treatments were: irrigation from berry softening (50 DAFB) to harvest to maintain the vines well-hydrated and no irrigation. Therefore, 15 vines were tagged for each of the following treatments:

- Leaf removal + no irrigation (LR + NI);
- Leaf removal + irrigation (LR + I);
- No leaf removal-control + no irrigation (C + NI);
- No leaf removal-control + irrigation (C + I).

Fruit Localization

Berry temperature was recorded by 48 thermocouples (12 per treatment) and their position was measured for the dataset, with the following coordinate system (Fig. 51): X indicates the height from the ground, Y indicates the distance from the trunk (from North-East to South/West direction) and Z the lateral distance from the center of the vines-row



Fig. 51. X, Y, Z coordinates used in the vineyard of cv. Sangiovese

Data regarding vines carrying the monitored berries, and berries localization themself are stored in the “1. Thermocouple position (2022) .xlsx” file, on the ATBCloud, presenting the data structure shown in Fig. 52.

node	thermocouple number	treatment	block	vine	canopy side	X height from the ground (cm)	Y distance from the trunk along the row in the South East direction (cm)	Z distance from the center of the row on the South-East side (cm)	Z distance from the center of the row on the North-West side (cm)	GPS	GPS coordinates taken 2 m above the ground
1	1	D + WS	1	2	SE	84	100	15		1	11.38303804 44.53583307
	2	D + WS		3	SE	88	41	7		2	11.38302819 44.5358263
	3	D + WS		2	NW	79	72		18	1	11.38303804 44.53583307
	4	D + WS		3	NW	80	44		13	2	11.38302819 44.5358263
	5	C + WS		1	SE	105	37	13		3	11.38298288 44.53579548
	6	C + WS		2	SE	80	65	15		4	11.38297271 44.53578866
	7	C + WS		1	NW	94	54		13	3	11.38298288 44.53579548
	8	C + WS		2	NW	85	98		7	4	11.38297271 44.53578866
...
...
...
6	1	D + WW	2	1	SE	97	40	8		15	11.38269518 44.53563991
	2	D + WW		1	SE	83	26	19		16	11.38267031 44.53562039
	3	D + WW		1	NW	94	60		15	15	11.38269518 44.53563991
	4	D + WW		4	NW	97	37		10	16	11.38267031 44.53562039
	5	C + WW		2	SE	92	88	8		17	11.38273241 44.53566405
	6	C + WW		2	SE	93	10	13		17	11.38273241 44.53566405
	7	C + WW		2	NW	88	51		19	17	11.38273241 44.53566405
	8	C + WW		2	NW	87	84		10	17	11.38273241 44.53566405

Fig. 52. Dataset 1 – thermocouple position data structure in 2022 season.

Sunburn symptoms evaluation

The sunburn damages (i.e., early sunburn damages, sunburn necrosis and berry shrivel) were visually estimated twice a week to monitor their occurrence and evolution (“3. Damage evolution of clusters with thermocouples (2022).xlsx” file on the ATBCloud – Fig.53). In all the clusters (N=24) with the thermocouples of vines on which leaf removal was performed (12 thermocouples in LR + I treatment and 12 in LR + NI) the sunburn damages appeared, while those clusters covered by the foliage (24) were not affected at all (C + I and C + NI treatments). The severity of the sunburn necrosis was reduced

by the irrigation and resulted higher on clusters on the South-East side of the canopy respect to those on the North-West side.

cluster	early sunburn damage										sunburn necrosis										berry shrivel											
	2 aug	5 aug	8 aug	11 aug	16 aug	23 aug	30 aug	6 sept	2 aug	5 aug	8 aug	11 aug	16 aug	23 aug	30 aug	6 sept	2 aug	5 aug	8 aug	11 aug	16 aug	23 aug	30 aug	6 sept	2 aug	5 aug	8 aug	11 aug	16 aug	23 aug	30 aug	6 sept
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	2	2	3	5	10	10	10	0	0	0	5	5	5	5	
2	5	10	15	0	0	0	0	0	0	0	0	0	0	10	15	15	20	25	0	0	0	5	5	5	10	10	10	10	10	10		
10	1	4	5	0	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	5	5	5	5	5	5	5	5	5	5	5	
.	
4	2	8	10	0	0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	5	5	5	5	5	5	5	5	5	5	5	
6	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	
11	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	5	5	5	
3	0	0.5	2	2	0	0	0	0	0	0	0	0	0	0	8	8	8	0	0	0	0	5	10	5	5	2	0	0	0	0	0	
6	15	18	25	0	0	0	0	0	0	0	0	0	0	20	40	45	45	45	0	0	0	5	10	5	5	5	5	5	5	5	5	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	6	0	0	0	0	2	2	3	3	3	3	3	3	3	3
15	0	1	1	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	1	2	0	0	0	0	0	0	0	0	0	2
2	2	8	10	0	0	0	0	0	0	0	0	0	0	5	5	5	10	10	0	0	0	5	10	15	5	5	5	5	5	5	5	
10	2	8	10	0	0	0	0	0	0	0	0	0	0	10	20	25	30	30	0	0	0	0	10	10	10	10	10	10	10	10	10	
12	0	2	2	5	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	5	10	10	10	10	10	10	10	10	10	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
1	1	8	10	0	0	0	0	0	0	0	0	0	0	2	7	10	10	20	0	0	0	0	10	10	10	5	5	5	5	5	5	
2	0	1	5	0	0	0	0	0	0	0	0	0	0	0	5	5	10	15	0	0	0	0	10	10	10	10	10	10	5	5	5	
11	2	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	10	10	10	10	10	10	5	5	5	
12	1	3	3	5	0	0	0	0	0	0	0	0	0	0	0	2	10	0	0	0	0	5	10	10	10	10	10	5	5	5	5	
3	2	5	5	0	0	0	0	0	0	0	0	0	0	0	2	5	10	10	0	0	0	0	5	10	10	5	5	5	5	5	5	
3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	5	10	10	10	10	10	5	5	5	
14	0	5	5	0	0	0	0	0	0	0	0	0	0	0	5	10	10	15	0	0	0	0	10	15	15	15	15	15	10	10	10	
14	0	5	5	0	0	0	0	0	0	0	0	0	0	0	5	5	10	15	0	0	0	0	15	15	15	15	15	15	10	10	5	

Fig. 53. Dataset of “Damage evolution of clusters with thermocouples” in 2022.

In Fig. 53 above is shown data structure of the sunburn symptoms evaluation: in add to all the information related to fruit localization (Fig. 52), here are presented the occurrence of sunburn symptoms, per each cluster in analysis, and per each timing during the season.

Fruit temperature and weather data

For all the presented datasets, seasonal meteorological data acquired by a weather station mounted in the vineyard were available and shared (“4. weather station (2022).xlsx” file on ATBCloud).

Fruit temperature data were collected following different methodologies as presented below in each of the dataset description.

Dataset 1 (Tcouple) - Continuous thermal measurements.

In this dataset, fruit temperatures were recorded by thermocouples from the day when leaf removal was applied (58 DAFB) to harvest. The same WSN and thermocouples used for apple fruits were utilized for monitoring berry temperature in the cv. Sangiovese vineyard.

We monitored the temperature of 12 berries (6 on the South-East side and 6 on the North-West side of the canopy) for each of the 4 treatments (N total = 48), from 29 July to 19 September 2022. Temperatures were recorded every 15 minutes but were reported as hourly average because unfortunately some data were missing. The thermocouples were inserted under the berry skin and twice a week was checked: at the first symptom of berry shrinkage, the thermocouple was moved into another berry, very closed to the previous (Fig. 54).



Fig. 54. Thermocouple inserted under the berry skin of a cv. Sangiovese berry

The dataset obtained in this way, was uploaded on the ATBCloud as “2. thermocouple data (2022).xlsx” file (Fig. 55)

block	Temperature SE side (°C)						Temperature NW side (°C)						Air Temperature inside the canopy (°C)		Air relative humidity (%)			
	I		II		III		I		II		III		I	III	I	III		
	node	2	2	6	6	4	4	2	2	6	6	4	4	3	4	19.35	19.45	90.9
termocouple number	1	2	1	2	1	2	3	4	3	4	3	4	3	4				
29/07/2022 00:00	19.1	16.9	18.0	18.0	18.4	18.0	18.9	17.2	18.7	18.1	18.1	17.1	19.35	19.45	90.9	91.5		
29/07/2022 01:00	18.5	16.4	17.5	17.5	17.9	17.4	18.4	16.6	18.1	17.5	17.7	16.6	18.93	19	92.2	92.83		
29/07/2022 02:00	18.3	16.5	17.4	17.3	17.7	17.3	18.6	16.4	18.0	17.4	17.5	16.5	19.05	18.7	92.88	95.3		
...
...
...
19/09/2022 10:02	25.6	26.7	23.3	23.2	23.6	32.2	21.7	20.0	21.7	19.7	24.1	20.7	22.15	23.3	53.13	49.75		
19/09/2022 11:02	27.1	28.5	23.5	23.5	23.4	37.2	22.6	20.8	22.5	21.6	23.4	21.5	23.17	24.4	52.48	50.15		
19/09/2022 12:02	28.7	29.7	25.5	25.4	25.0	31.0	23.6	21.9	23.2	22.8	23.1	22.5	24.12	25.47	51.72	47.85		
19/09/2022 13:02	28.3	25.0	24.2	24.2	25.0	26.8	24.8	22.7	23.9	23.7	23.8	23.2	24.8	26.37	49.68	45.33		

Fig. 55. Tcouple dataset structure.

In Fig. 55, is presented the data structure per each of the treatment’s sheet in the file: Celsius degree temperatures data are presented hourly per each *block*node*Termocouple number*, both for *SE* and *NW* canopy sides; in the dataset are reported also ‘*Air temperature inside the canopy*’ and ‘*Air relative humidity*’.

Dataset 2 (RGB-D/T-system) – Fruit thermal scanning dataset – PRELIMINARY

This preliminary dataset was extracted exploiting the same identical process and RGB-D/T system of the ones related to apple fruits presented in the previous chapter.

For grape cluster the vines analyzed throughout the RGB-D/T system were the same monitored with the thermocouples and so included in “Dataset 1”. Field data collection protocol is reported in “Field data collection with the RGB-D/T system” subchapter (pag. 30). For this dataset the utilized *X*, *Y*, *Z* coordinate system is different from the one presented for “Dataset 1”, being the same used for apple datasets (i.e., Fig. 43 and not Fig. 51).

In total, 7 out of 15 (+1 unhealthy vine that was discarded) vine images were correctly analyzed, obtaining thermal and positional information for a total of 29 grape clusters. Most of the missing images were not analyzed due to leaf occlusion level present (due to the treatment condition – Fig. 56) that made impossible the grape cluster detection for the system; grape cluster detection not falling in the thermal camera field of view and/or missed grape cluster detection could be happened, reducing the number of cluster’s temperature and position information collected.



C

LR

Fig. 55. Occlusion level of different treatments: on the left the control treatment (C) with no leaf removal; on the right treatment 'Leaf Removal' (LR) with leaf removal favoring cluster detection. Leaf occlusion impede clusters detection

In this case, reference data for the detected and analyzed clusters were collected for temperatures. Cluster position was collected but due to the small dimension of the tag in the collected images (Fig. 55 - right), it was not possible to properly evaluate fruit position errors of the system. Contrarily it was possible to evaluate system performances for temperature estimation: these were already presented in previous chapter "Fruit temperature extraction from RGB-D/T-system collected images – Evaluation" (pag. 41) and resulted in a RMSE / mean error ranging from ± 3.43 / -0.96 °C to ± 10.36 / -9.79 °C, for single cluster temperature measured with thermocouples (n=16) while considering the mean fruit temperature extracted for one side of the plant (i.e., the mean of all detected clusters) and comparing it to thermocouples measurements for those plant side (n=6), the RMSE / mean error range from ± 3.42 / -1.83 °C to ± 12.20 / -11.97 °C.

The obtained dataset was uploaded as "RGBD-T-System_GRAPE_Pos&Temp_Dataset.xlsx" file, on the ATBCloud. The dataset includes only the information of those fruits correctly detected and that passed all the process filtering steps explained in "Deliverable No D2.2 – Continuation" (pag. 25). The dataset is structured in seven different sections (sec0 to sec6), as presented in "Dataset 3" of UNIBO Apple fruit data temperature and position dataset, with the same identical data. Also in this case, a more in-detail description is present inside the dataset file in the 'readme' and 'REFERENCE crs' sheets.

A graphical representation of the collected data is presented in Fig. 56 reported below. In this case it can be noticed a temperature estimation error, where a grape cluster seem to be at > 80 - 90 °C.

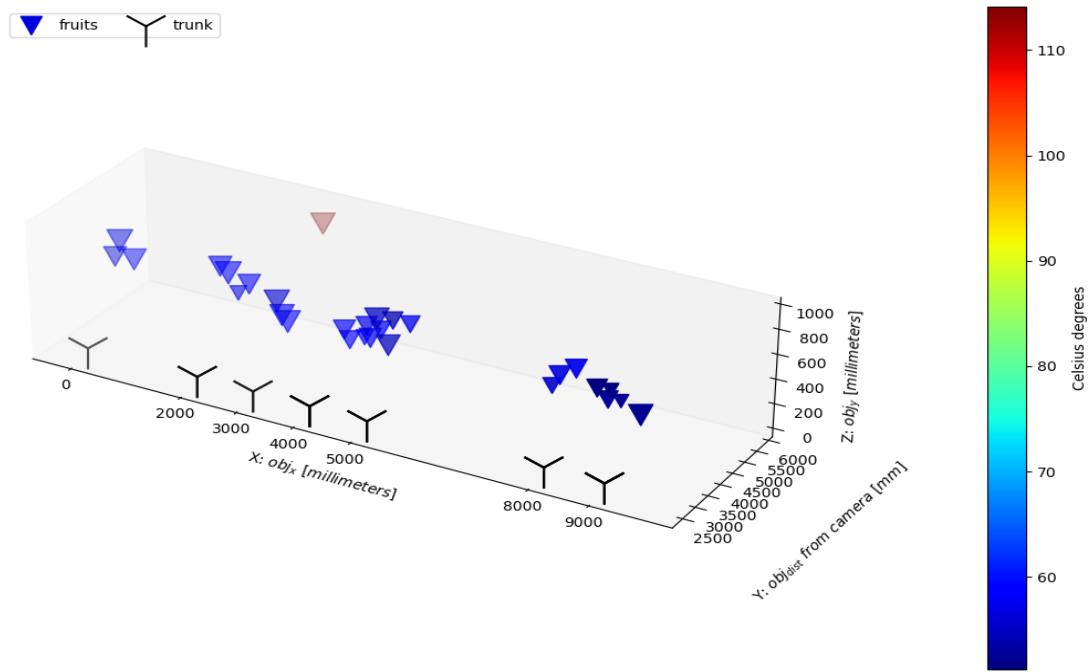


Fig. 56. Copy of Fig. 39 -Grape of the previous chapter, representing grape cluster temperature 3D representation at “orchard level”, with fruit position relative to a defined orchard origin, and temperature represented by color-scale.

Considering the preliminary testing of the platform performances (reported above), due to the current possible high error of the system (up to ± 10.36 °C and ± 0.15 m), we suggest avoiding using data from this dataset to train model for investigating sunburn dynamics. Indeed, the error from Fig. 56 suggests to add a filtering step to discard “not realistic” temperature values before using these for further analysis.

However, this preliminary dataset is presented, and shared, so to have feedbacks for possible improvements and to make other partners able to evaluate it.

ATB- Apples segmented temperature

Data acquisition

The measurement was carried out using the linear conveyor system moved at 20 mm s^{-1} ($\pm 0.05 \text{ mm}$ accuracy) forward speed. A mobile 2D LiDAR sensor (LMS-511, Sick AG, Waldkirch, Germany) was mounted vertically on the metal frame at 0.7 m above the ground level. The LiDAR sensor configured with a 0.1667° angular resolution, 25 Hz scanning frequency, a scanning angle of 180 and a wavelength of 905 nm. Additionally, a thermal camera (A655sc, FLIR Systems Inc., MA, USA) placed at 0.2 m distance above the laser scanner. The camera has a spatial resolution of 640×480 pixels at 50 Hz, with a spectral range from 7.5 to 14 μm , an operational temperature range from -40°C to 150°C and a thermal resolution $< 0.05^\circ\text{C}$. A lens (T198065, FLIR Systems Inc., MA, USA) with a focal length of 6.5 mm (diagonal 80°) is attached to the camera.

A FasTRAK 3D digitizer (Polhemus, Colchester, VT, USA) was used to manually digitize the position of apples and cherries on trees with an RMS accuracy of 0.2 cm. The FasTRAK sensor connects via cable to a G4 ‘Hub’ module; this module transmits digitized data via Radio Frequency (RF) to an RF module connected to the computer via USB. A ‘Source’ module generates the electromagnetic field required to track the sensor. In parallel, an infrared thermometer was used to manually acquire the temperature on apple ($n = 84$) and cherry ($n = 54$) surface (TempRef) from the two sides in apple trees

($n = 7$) and from the east side in cherry trees ($n = 9$). The TempRef from each apple and cherry was compared with the extracted men temperature by means of LiDAR point cloud (TempLiDAR) from both fruit. The fruit point clouds were manually segmented in CloudCompare (2.10, GPL software, Paris, France). The measurements carried out at 40, 70, 85, 115 and 135 days after full bloom (DAFB) in apples. Whereas in cherries, the measurements carried out at 55, 65, 75 and 80 days after full bloom (DAFB). The dataset of each date saved in a .csv file, while it was consisted of five columns including the manually measured digitiser position (X,Y,Z) of apples, the TempRef and TempLiDAR.

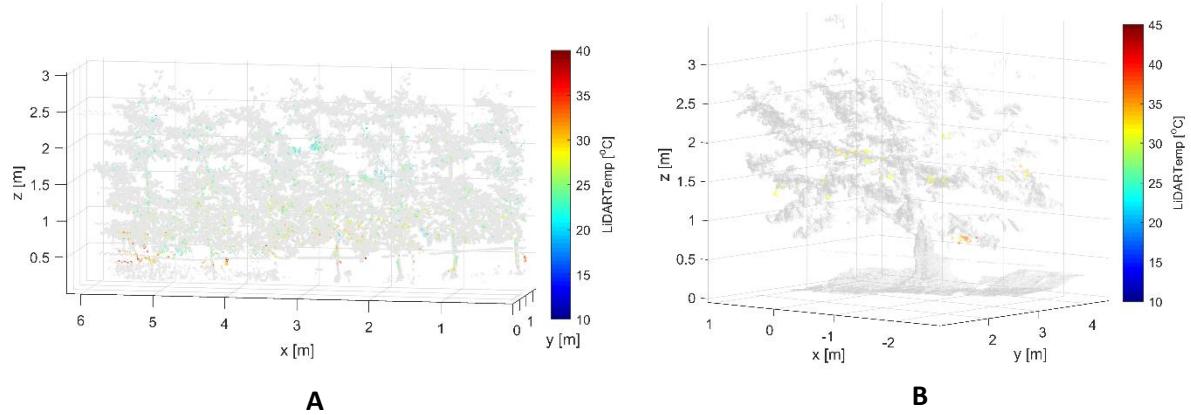


Fig. 57. 3D point clouds of A. 7 apple trees and B. cherry tree segmented from 3D point cloud of complete tree row and presented in newly built coordinates with the segmented fruit temperature labelled in red, measured on the last measuring date at harvest.

Preliminary results

After fruit localization, the extracted fruit temperature was compared to the manually measured diameter during the growth period (Table 6). The TempLiDAR was related to the manual measurements, especially in the DAFB40 and DAFB135, resulting in $R^2 = 0.89$ and $R^2 = 0.91$ with RMSE = 1.71 % and 0.78 %, respectively. Generally, enhanced measuring uncertainty was noticed on DAFB70 and DAFB85. More specifically, a less pronounced relation was observed on this measuring dates, presenting an RMSE of 2.58 and 253 % at DAFB70 and DAFB85, respectively (Table 6). Similar good results were reported in cherries presenting low and moderate correlations over fruit development with highest coefficients of determination at DAFB75 ($R^2 = 0.76$, RMSE = 2.49 %) and DAFB80 ($R^2 = 0.74$, RMSE = 2.53 %). The visually higher canopy density in lower tree section enhanced the error. Nevertheless, the LiDAR based analysis provides an accurate tool for fruit counting.

Table 6. Reference data and LiDAR derived estimations with error analysis (root mean squared error, RMSE; coefficient of determination, R^2) considering temperature on fruit surface (LiDARTemp) measured at the tree in five and four growth stages of apples and cherries in day after full bloom (DAFB).

		DAFB40	DAFB70	DAFB85	DAFB115	DAFB135
LiDARTem p in apples (°C)	Mean	25.53	28.16	28.29	22.82	15.29
	RMSE (%)	1.71	2.56	2.53	1.76	0.78
	R^2	0.89	0.65	0.76	0.80	0.91
LiDARTem p in Cherries (°C)		DAFB55	DAFB65	DAFB75	DAFB80	X
	Mean	24.32	27.65	23.67	21.34	
	RMSE (%)	2.34	2.78	2.49	2.53	
		0.71	0.68	0.76	0.74	

The linear model was used to express the overall relationship between NDVIRef and NDVILiDAR, revealing high $R^2 = 0.90$, RMSE = 2.08 % calculated from the 3D point cloud of apples measured in the field (Fig. 58). In parallel, the overall relationship between TempRef and TempLiDAR in cherries was expressed also with a linear equation, revealing $R^2 = 0.91$, RMSE = 1.71 % considering all measuring dates (Fig. 58)

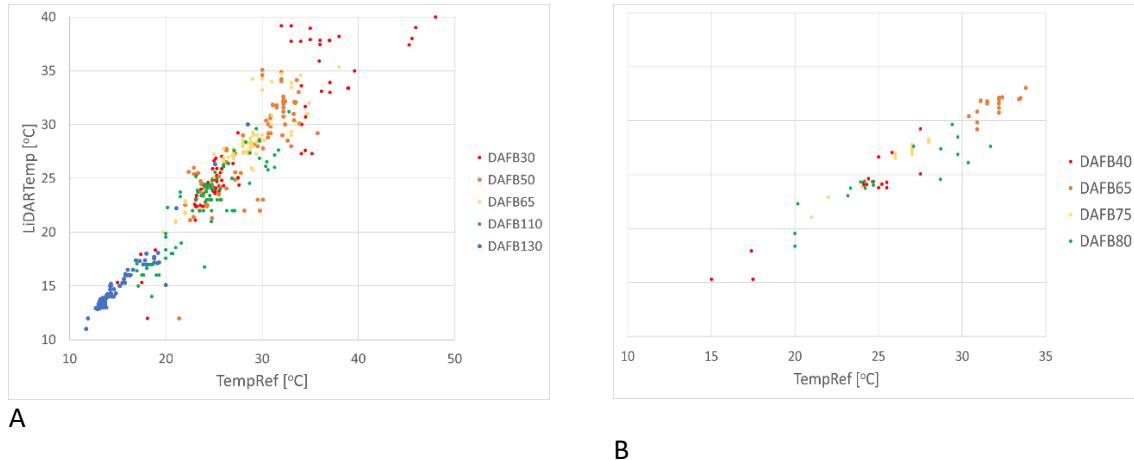


Fig. 58: Scatterplots of LiDAR derived fruit temperature (TempLiDAR) measured with manually measured TempRef in A. apples ($n = 84$) and B. cherries ($n = 54$) over the measuring dates during fruit development of

References

- Hackel, T.; Wegner, J.D.; Schindler, K. Contour detection in unstructured 3D point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1610–1618.
- Kooi, B. MTi User Manual, MTi 10-Series and MTi 100-Series. *Document MT0605P.B.* 2016, pp 35–36.
- Lin, C.H.; Chen, J.Y.; Su, P.L.; Chen, C.H. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. *ISPRS J. Photogramm. Remote Sens.* **2014**, *94*, 70–79.
- Tsoulias, N.; Paraforos, D. S.; Fountas, S.; Zude-Sasse, M. Estimating Canopy Parameters Based on the Stem Position in Apple Trees Using a 2D Lidar. *Agronomy* 2019. <https://doi.org/10.3390/agronomy9110740>.
- Vázquez-Arellano, M.; Reiser, D.; Paraforos, D. S.; Garrido-Izard, M.; Burce, M. E. C.; Griepentrog, H. W. 3-D Reconstruction of Maize Plants Using a Time-of-Flight Camera. *Comput. Electron. Agric.* 2018, *145* (August 2017), 235–247. <https://doi.org/10.1016/j.compag.2018.01.002>.



