Mirko Piani, Gianmarco Bortolotti, Alexandra Boini

Department of Agricultural and Food Sciences - University of Bologna (IT)

SHEET DELIVERABLE 2.2 – CONTINUATION –

Fruit (Apple and grape cluster) Thermal and Positional data extraction by means of RGB-D/Thermal cameras, neural networks and computer vision

Deliverable Description

Deliverable hypothesis

Deliverable No	Hypothesis
D2.2	• 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	• 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 13-17). 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) (Figure 2). 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.

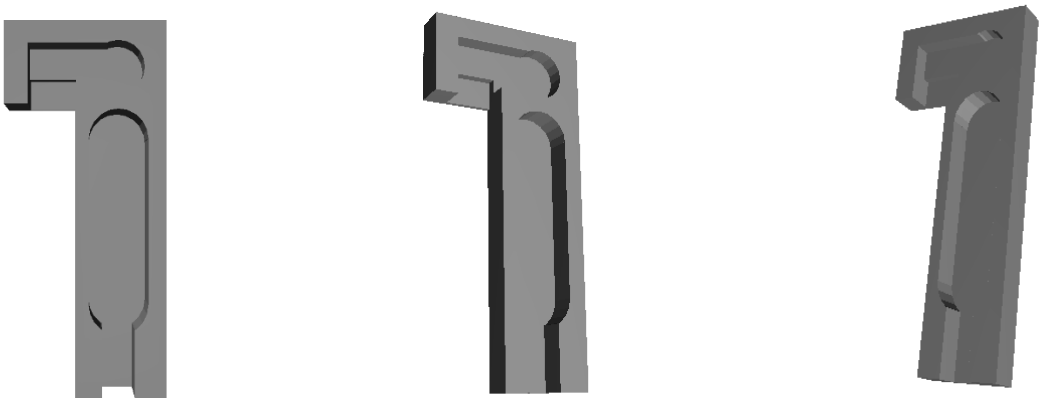

Figure 1. representation of the 3D printed case holding the thermal and RGB-D cameras

In order to firmly hold the cameras during the measurements, a 3D printed case was crated and customized to fit with sensors and cables (Figure 1, Figure 2). 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° x 42°; Thermal camera: 32° x 32°), 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.

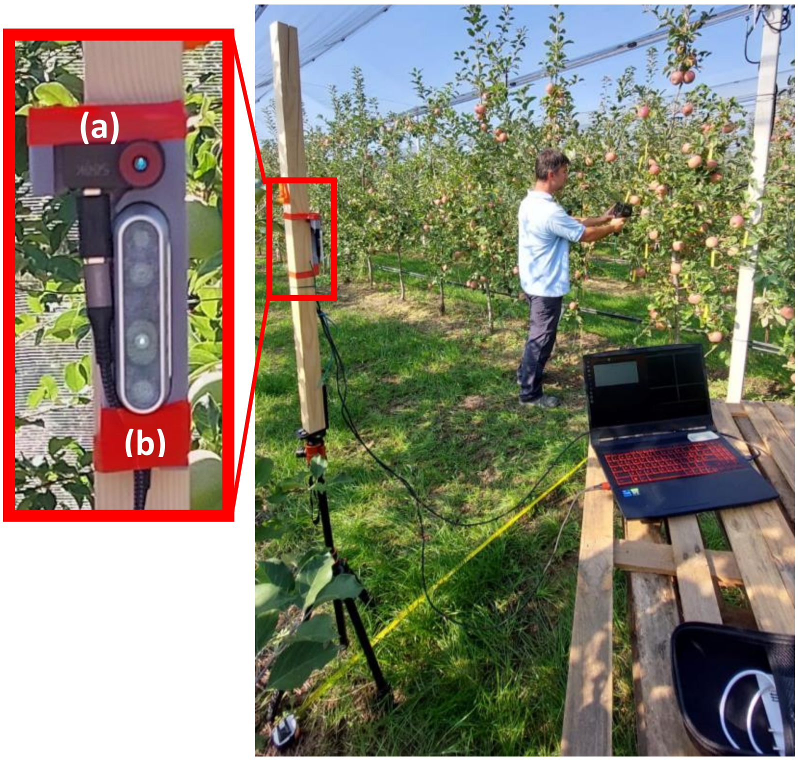

Figure 2. sensors setup: upper camera is SEEK thermal Compact Pro camera (a), while lower camera is IntelRealsense D435 (b)

RGB and Thermal image alignment

ROS-extracted images were either matrices like 1920 x 1080 x 3 or 1280 x 720 x 3 in uint8 datatype for RGB-D color images, while thermal ROS-extracted images are 320 x 240 x 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 (Figure 3). From images of this panel, an alignment process was performed as follows.

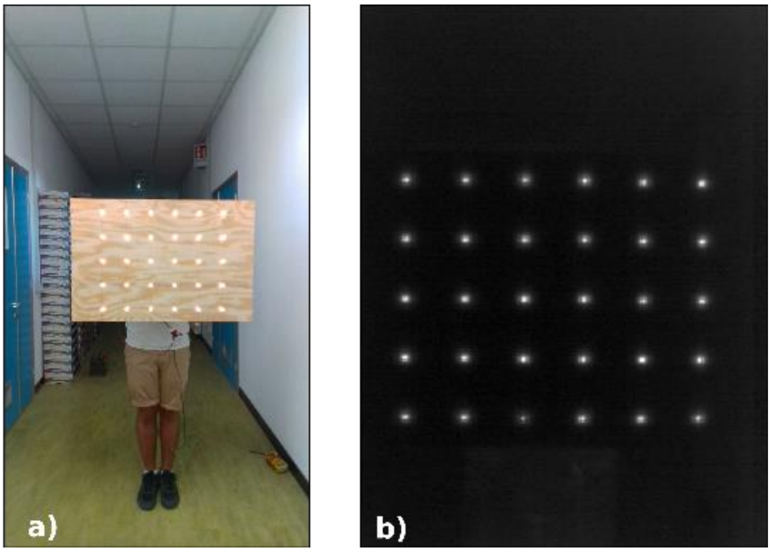

Figure 3. alignment panel images collected at 2.6m approx. by a) RGB-D camera – res: 1920 x 1080; b) Thermal camera (normalized data) -res: 320 x 240.

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 1).

Table 1- 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
detector= cv2.SimpleBlobDetector_create(params)				# keypoints

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.

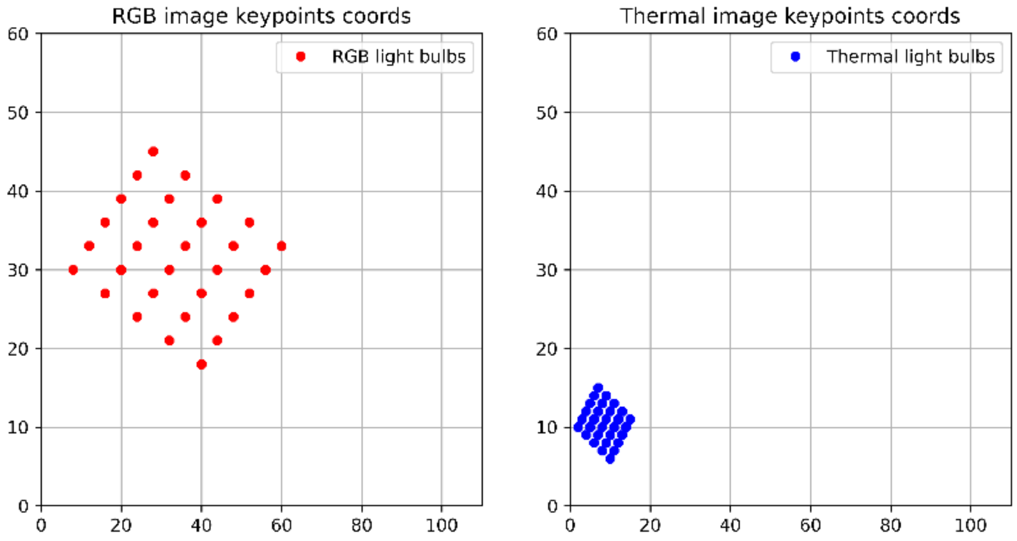

Figure 4. Graphical representation of the difference in RGB and thermal keypoints scale for a pair ROS-extractedof synchronized images.

Then, for all the pairs of synchronized RGB-Thermal images (both presenting 30 filtered blobs coordinates - Figure 4), 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** (*xmin, ymin*), **P_2** (*xmin, ymax*), **P_3** (*xmax, ymax*) and **P_4** (*xmax, ymin*), those represent the minimum bounding box enclosing all the blobs identified (Figure 5-a). Following, the four points related to the thermal image only are projected into the RGB canva (Figure 5-b) thanks to scaling factors (**SFx** and **SFy**) obtained as follows:

$$\mathbf{SFx} = (RGB\_P_4\_x - RGB\_P_1\_x) / (THERM\_P_4\_x - THERM\ \_P_1\_x)$$
$$\mathbf{SFy} = (RGB\_P_2\_y - RGB\_P_1\_y) / (THERM\ \_P_2\_y - THERM\ \_P_1\_y)$$

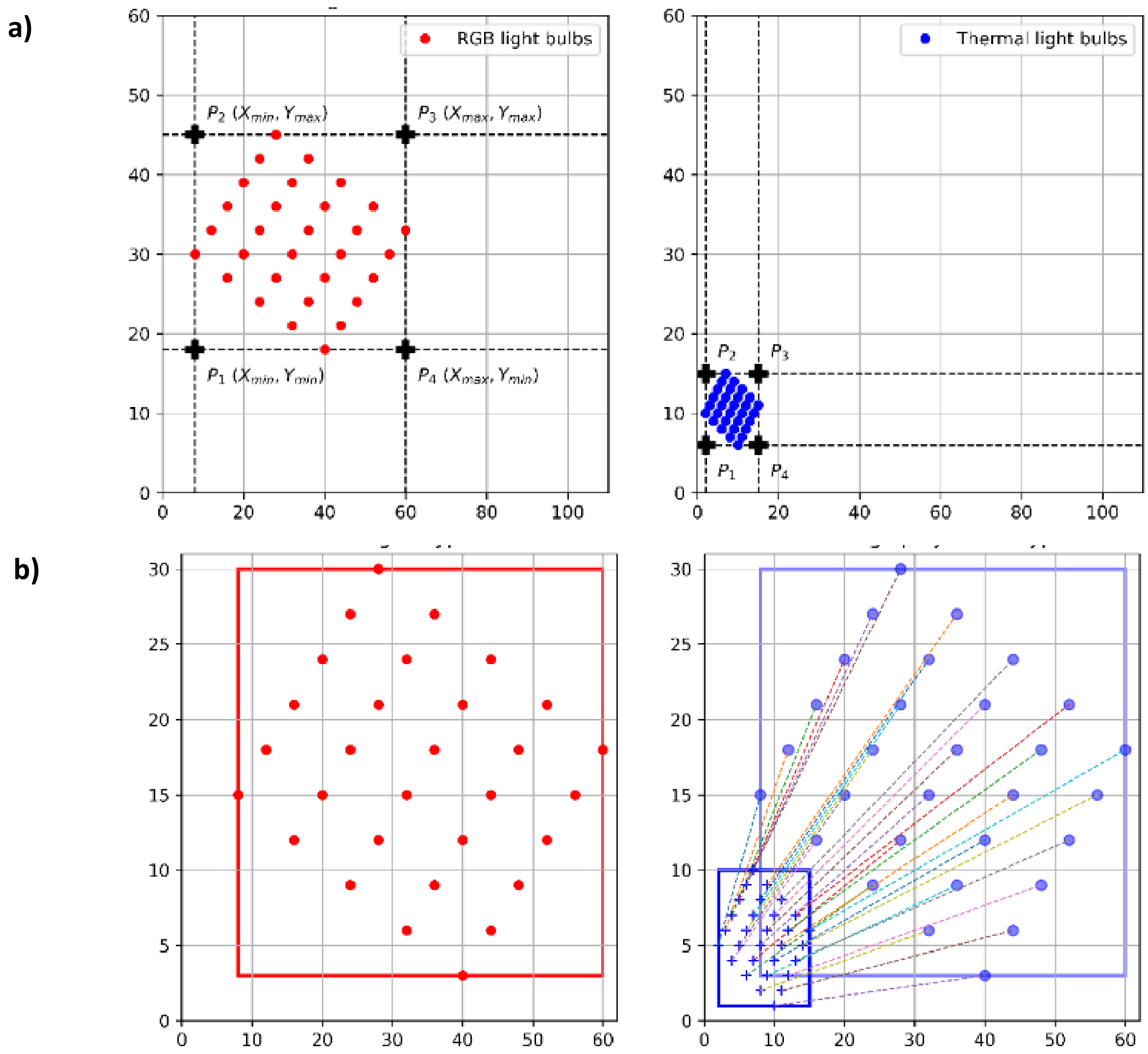

Figure 5. a) representation of the keypoints analysis (RED = RGB, BLUE = THERMAL) to define the four alignment points (P_1, P_2, P_3, P_4); b) representation of thermal keypoints' projections to match with RGB keypoints' positions

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 (Figure 6-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.

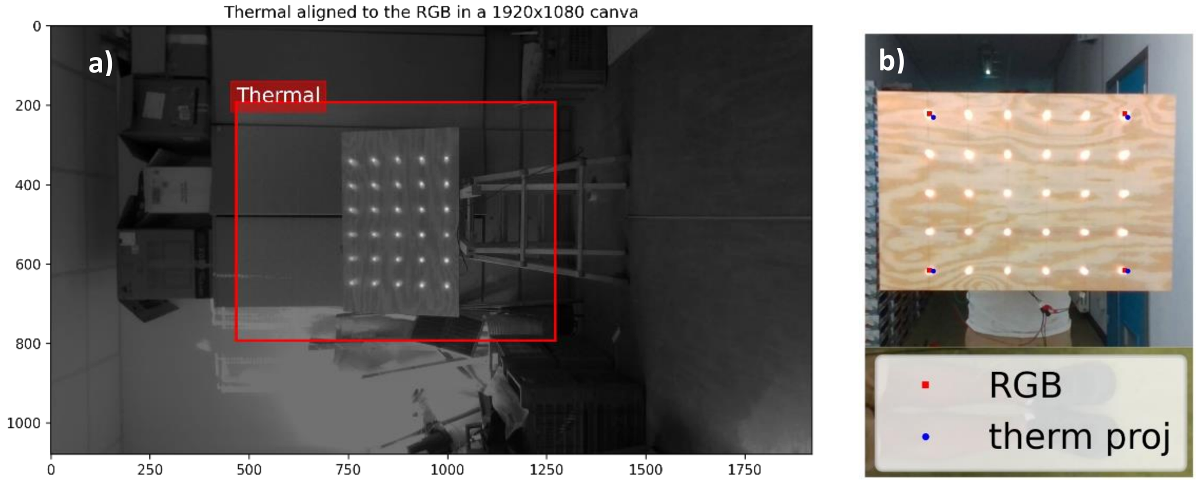

Figure 6. a) grey scale RGB image of the alignment panel containing the aligned thermal image at 1920 x 1080 resolution, in this case: the slight difference in lightbulbs lightness is due to the alpha merging factor between the images and the alignment errors. b) representation of the error in thermal-to-RGB 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 Figure 6-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.

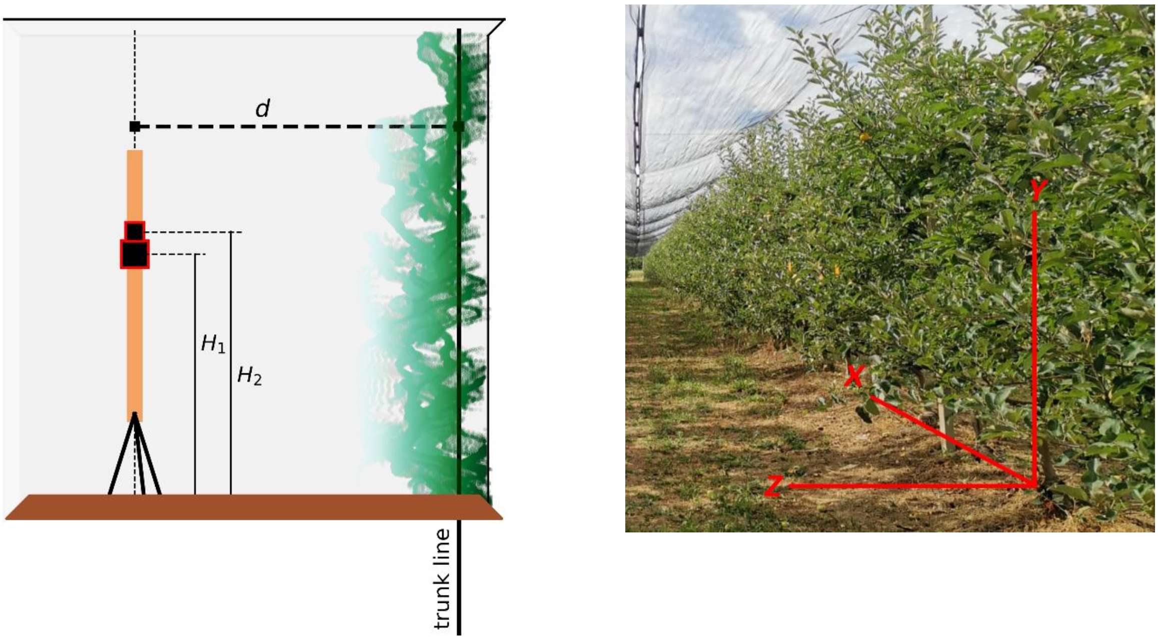

Figure 7. Cameras positioning during the field data collection. Cameras are placed parallel to the tree row plane at a "d" distance. Field data collection height was measured in the middle point between H_1 and H_2 (i.e., center camera height from the ground).

Apple data collection protocol

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 - Figure 7), 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 - Figure 7). 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 and 6 trees presenting green fruit 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 protocol

For grape data collection, a similar set up using the same *X, Y, Z* dimension and approach was utilized (Figure 7). 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 13-17), 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 (one for apples and one for grape clusters) 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 (Figure 13,Figure 20).

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, Huawaei 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 https://play.google.com/store/apps/details?id=com.tyriansystems.SeekThermal&hl=it&gl=US&pli=1) 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 Figure 8, 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): Figure 8-a shows three clear raw thermal data domains in which relationship change (<3000, 3000-7000, >7000), while Figure 8-b shows the detail of the domain 3000-7000, where the average linear regression seems to fit good with the presented thermal domain.

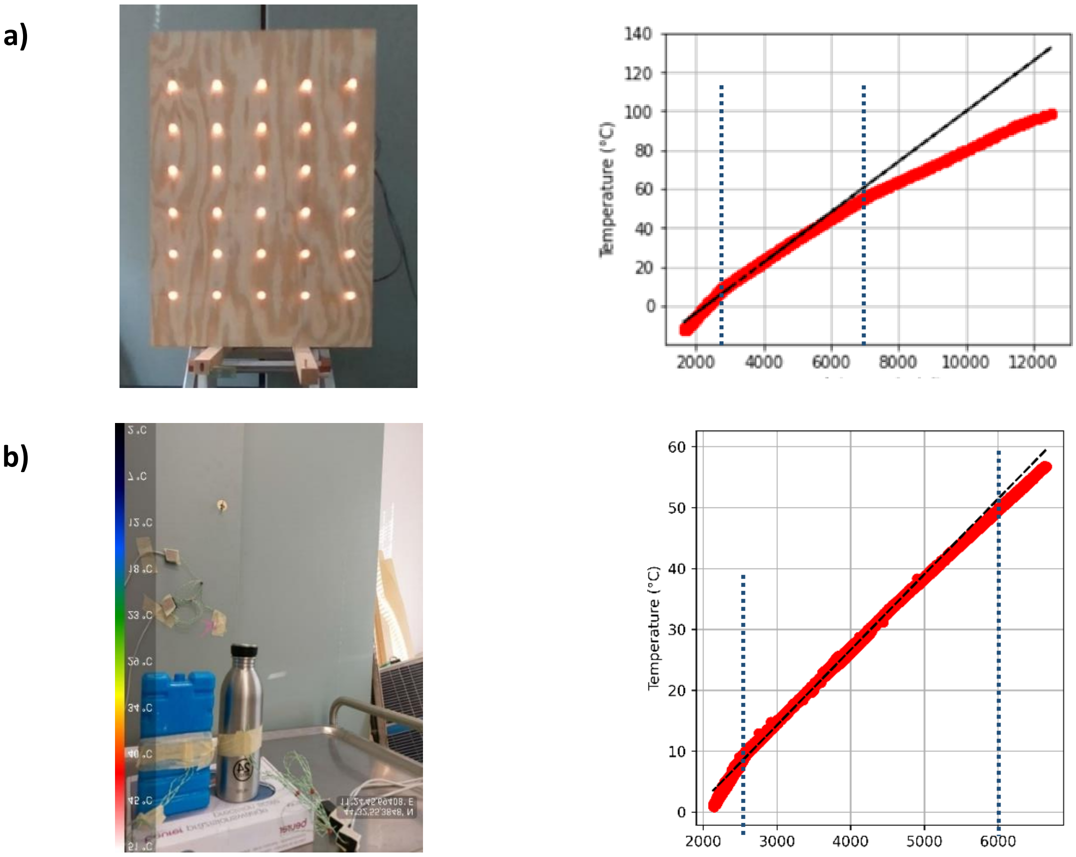

Figure 8. Thermal calibration set up (left) and related plots of raw thermal vs Celsius degrees obtained through the official SEEK smartphone (right); red dots represent the plotted data, black line is the general linear regression. ***a)*** *Thermal calibration set up exploiting the alignment panel 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 Figure 8), 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 Figure 9-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 x 0.1 m: refrigerated container, background wall area). On the other hand, in Figure 9-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 Figure 10 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).

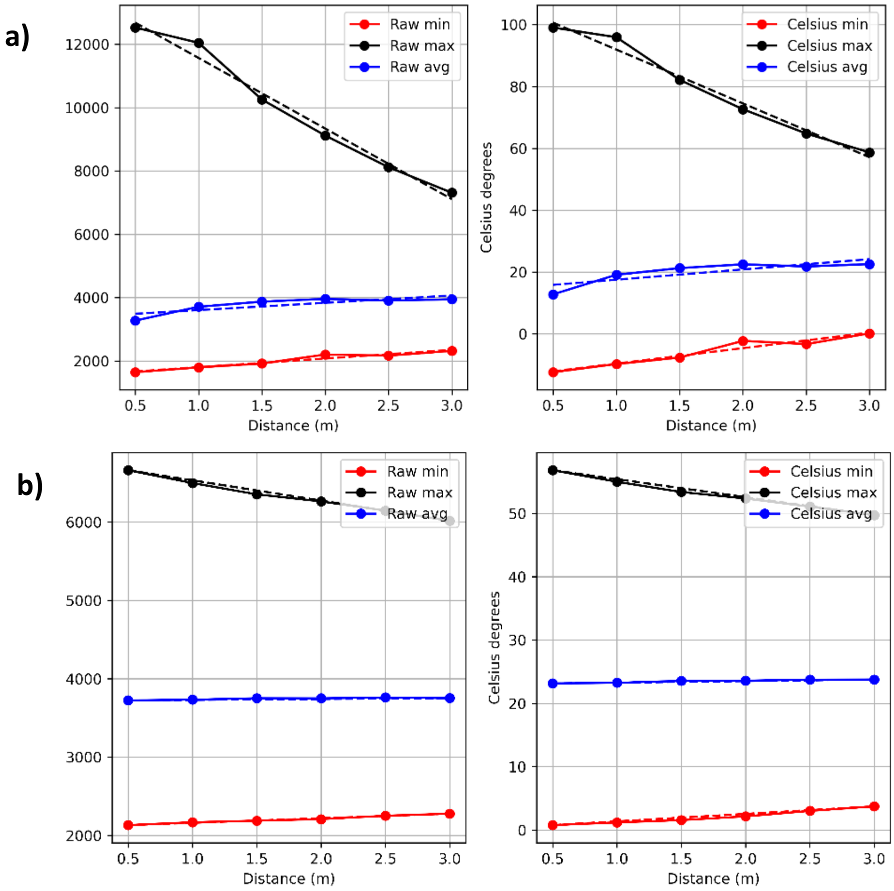

Figure 9. Effect of distance on the temperature estimation of the same object. From left to right are shown raw thermal data and Celsius degree data for 'max', 'avg' and 'min' that are respectively maximum, ambient and minimum temperatures in the scene; these are represented respectively in ***a)*** *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 (Figure 8-a and b), it was opted to utilize the optimal linear functions coefficient obtained by this set up (Table 2).

Table 2. 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	-28.62827940000000000
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 assume 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 Figure 10 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 3).

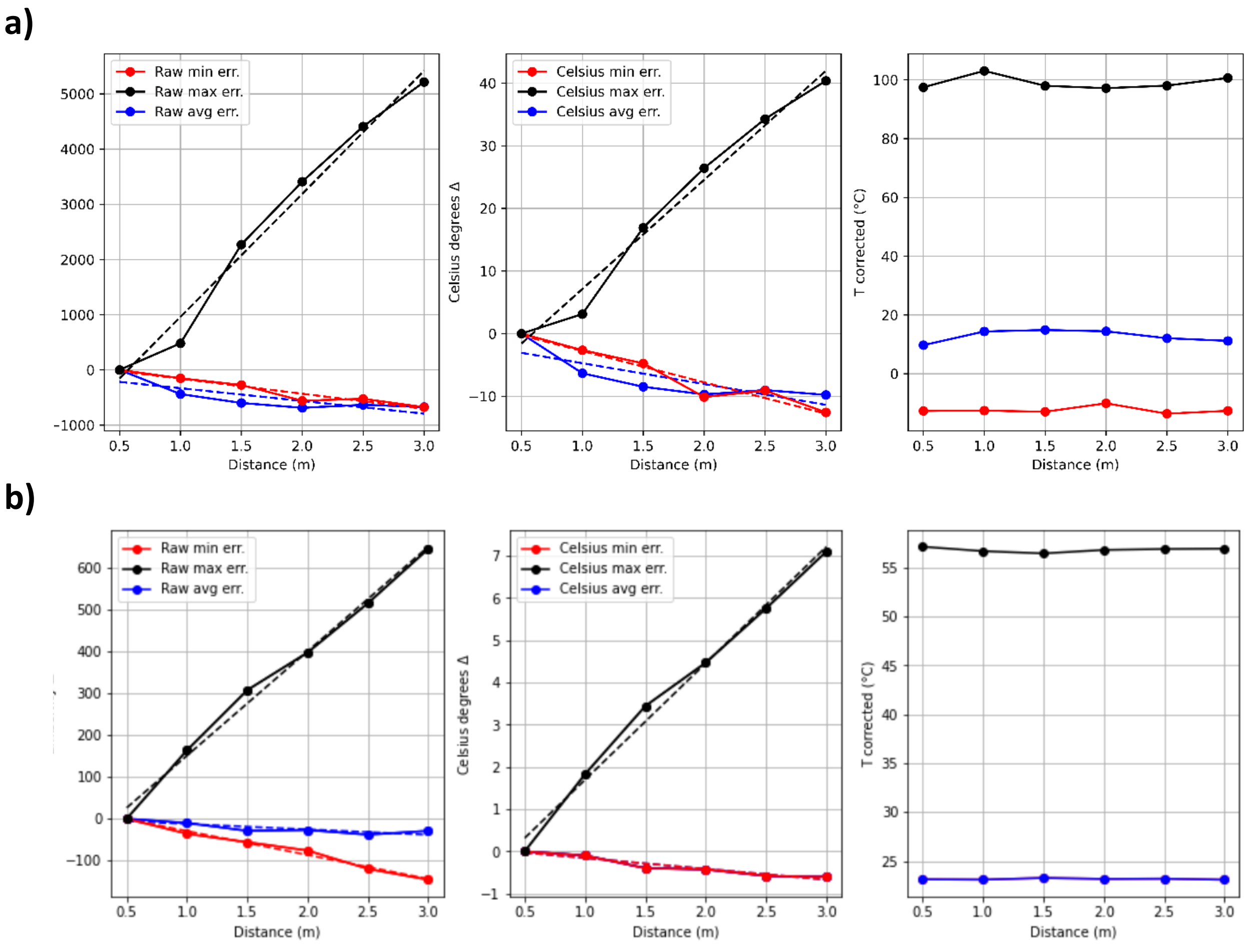

Figure 10. Temperatures errors related to distance of collection in raw (left) and °Celsius format (central); dashed lines represent linear regression through the points. On the right, the temperature values after the application of the correction equation. a) refer to lightbulbs set up; b) refer to 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 3 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 3 – 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 cm^2), 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 Figure 11 below.

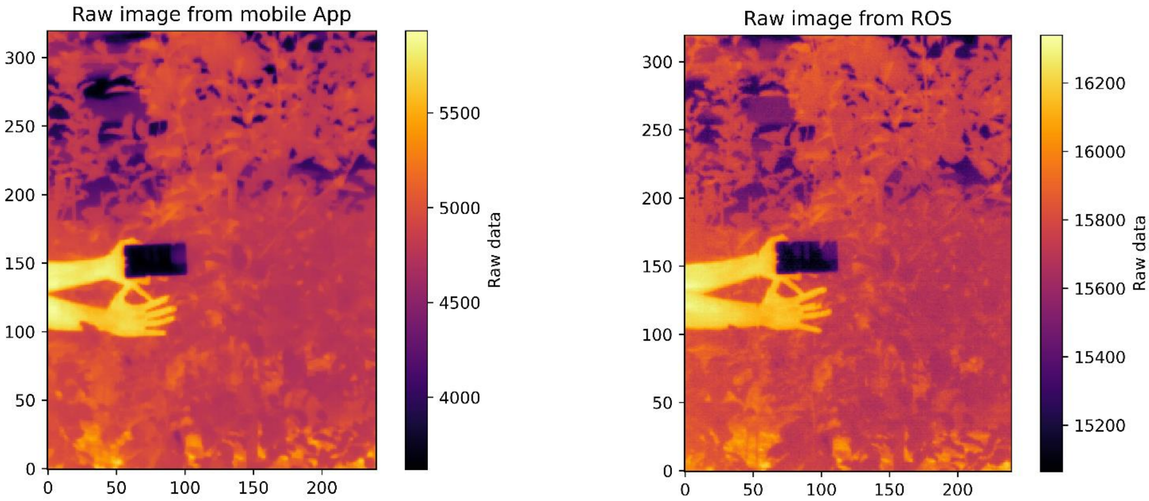

Figure 11. Comparison of scale values between SEEK app and ROS -collected raw thermal data for the same identical scene

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 2). 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 (Figure 12), 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 4 below.

Table 4. 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

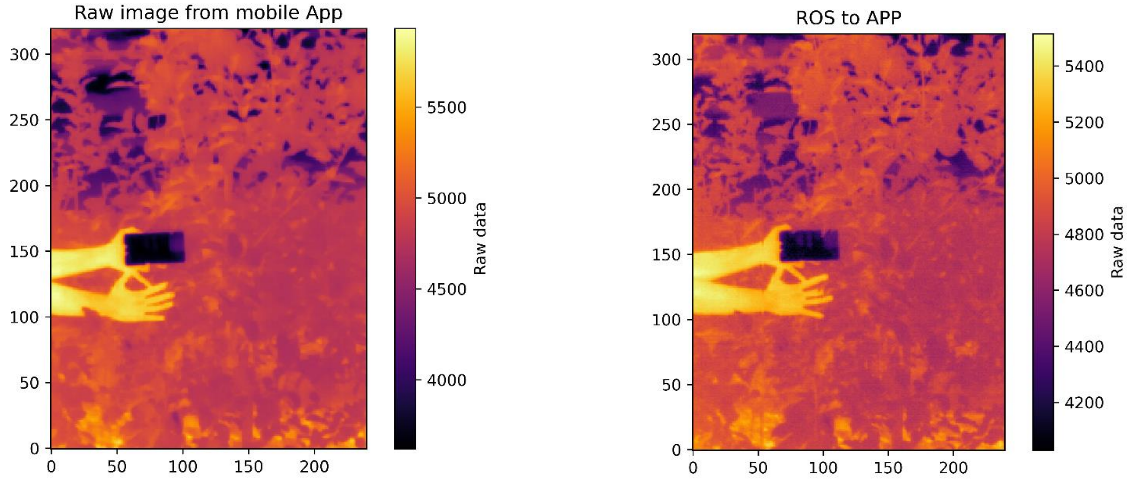

Figure 12. Original SEEK app data (right) and result of the ROS_to_APP function application to ROS-collected thermal data (right). Is highlighted the reduced, but proper scale range, and the good representation of the thermal variability after the conversion.

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.

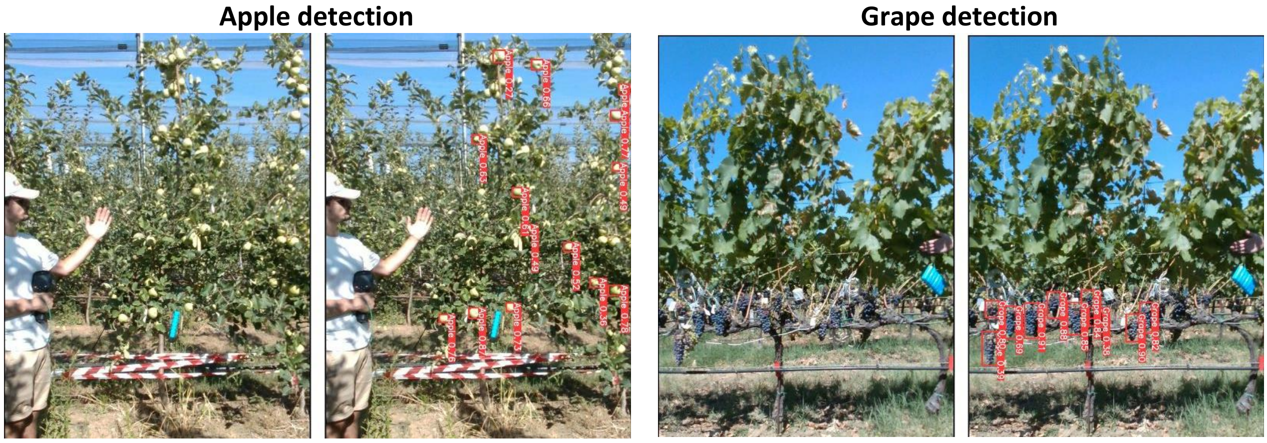

Figure 13. YOLOv5 trained model application for fruit detection: fruit detected are represented by red rectangle (bbox)

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 (Figure 14).

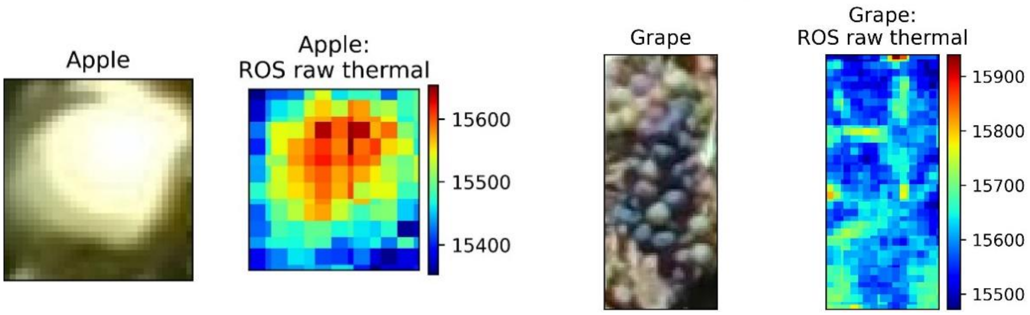

Figure 14. Examples of RGB (left) and relative aligned ROS-raw thermal data (right) bounding boxes

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 Figure 15: 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.

Reliable thermal data in the fruit bbox (Tbbox mean > 14500) **-proceeed to next step-**	**Non reliable thermal data in the fruit bbox** (Tbbox mean < 14500) **-discard the detected fruit-**

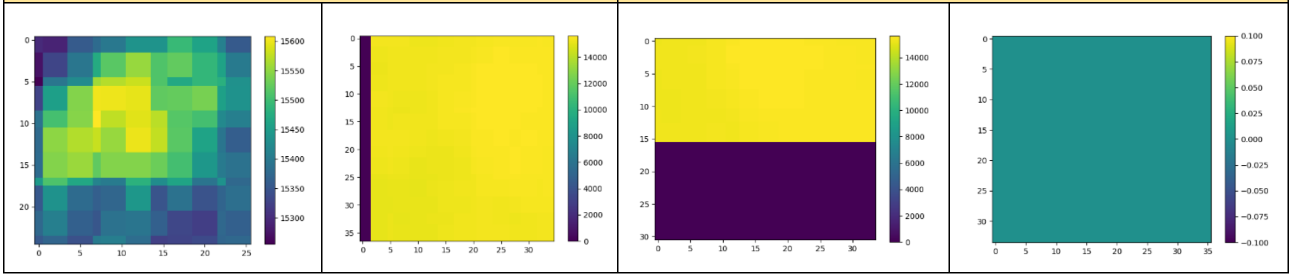

Figure 15. Representation of filtering step for defining reliable/non reliable thermal data contained in a detected fruit bbox.

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 (Figure 16). 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.).

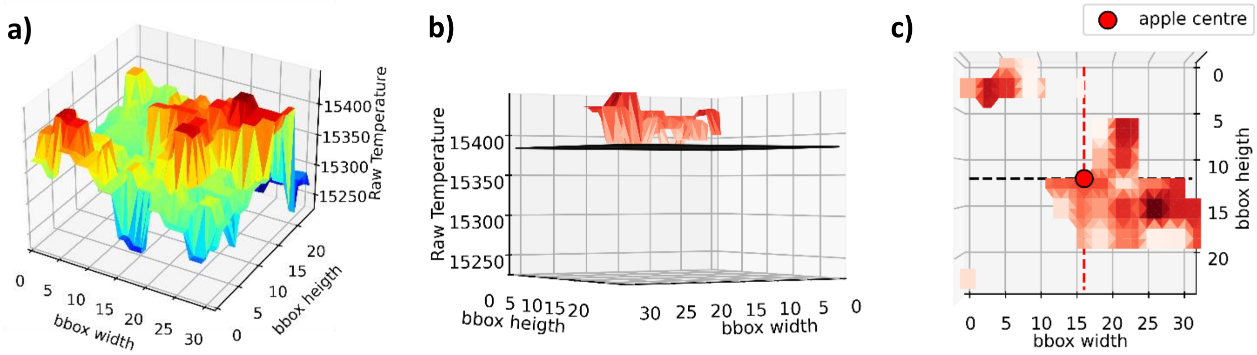

Figure 16. a) raw data thermal profile of the Tbbox; b) quantile filtering removing data <70th perc.; c) 70th perc. filtering result

5- After this quantile filtering step only raw thermal data of the 70th – 100th percentile remain for the following conversion calculation and extraction:

a. first raw thermal data are converted in their official seek app values range using the previously presented conversion function (Table 4);

b. then according to the obtained value and shooting distance the optimal raw-to-Celsius conversion function is applied (Table 2).

c. At lats, the minimum, mean and maximum temperature of the spot are computed and corrected for the distance error (Figure 10). 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 3.

d. 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 Figure 17 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.

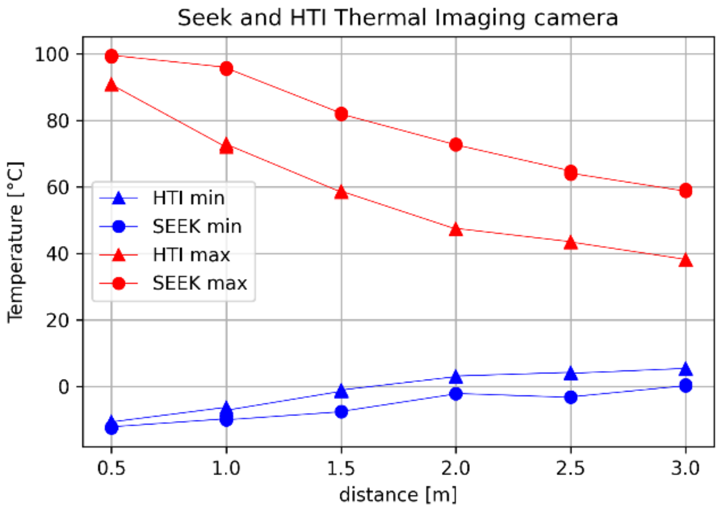

Figure 17. Comparison between SEEK and HTI temperature measurements at different distance. Each point represents the mean of three replicates.

Then a comparison based on field data collection was done. In this case data collection occurred following the presented apple data collection protocol (Figure 7), 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 Figure 18, 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) (Figure 18), this because the center of the scene was always and only framing the target objects.

	Tmin_SEEK	**Tavg_SEEK**	**Tmax_SEEK**	**Tpoint_HTI**	**Tmin_HTI**	**Tmax_HTI**
Tmin_SEEK	1					
Tavg_SEEK	0.994369708	1				
Tmax_SEEK	0.967724864	0.986328732	1			
Tpoint_HTI	**0.976161513**	**0.965340494**	**0.932539036**	1		
Tmin_HTI	0.488047637	0.48311038	0.483338432	0.525161596	1	
Tmax_HTI	0.062430662	0.086712329	0.161931994	0.078561197	0.282938777	1

Figure 18. 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. Despite this, the presented preliminary results are quite interesting considering the strong correlation between HTI point and all the ROS-extracted SEEK temperatures.

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 (Figure 19-a and b).

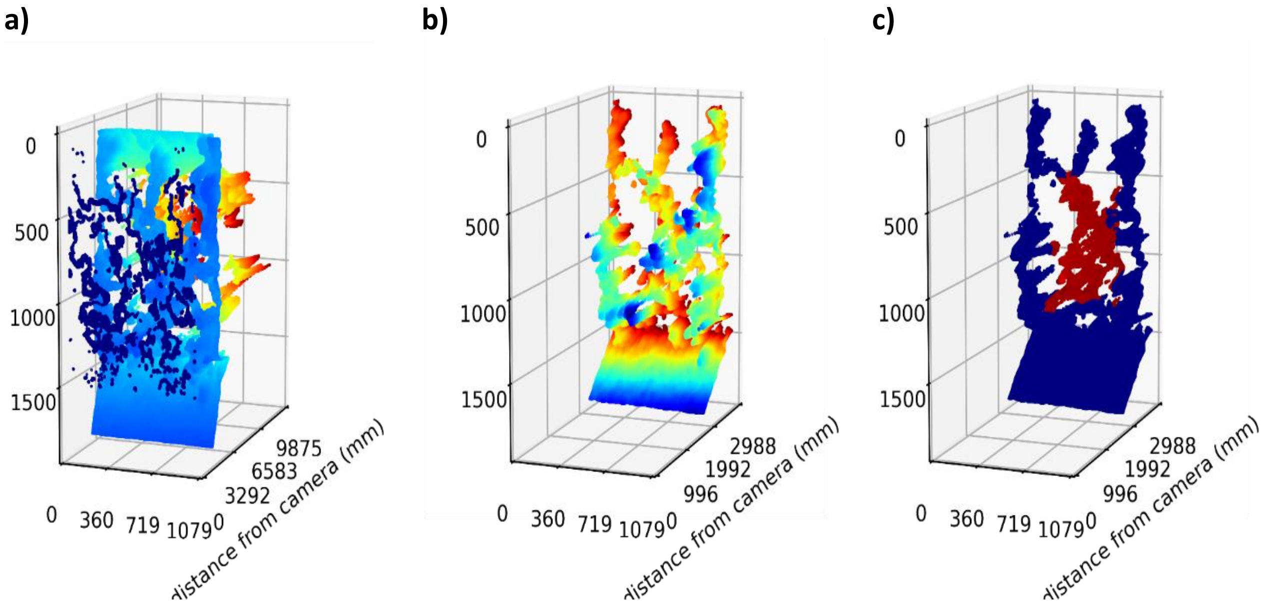

Figure 19. Representation of an apple tree depth map in a 3D plot; X and Y dimensions represent the pixel coordinates at which depth information (Z dimension – distance from the camera) is stored. ***a****) Unfiltered depth map for null values (i.e., dark blue points).* ***b****) Filtered depth map for 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).

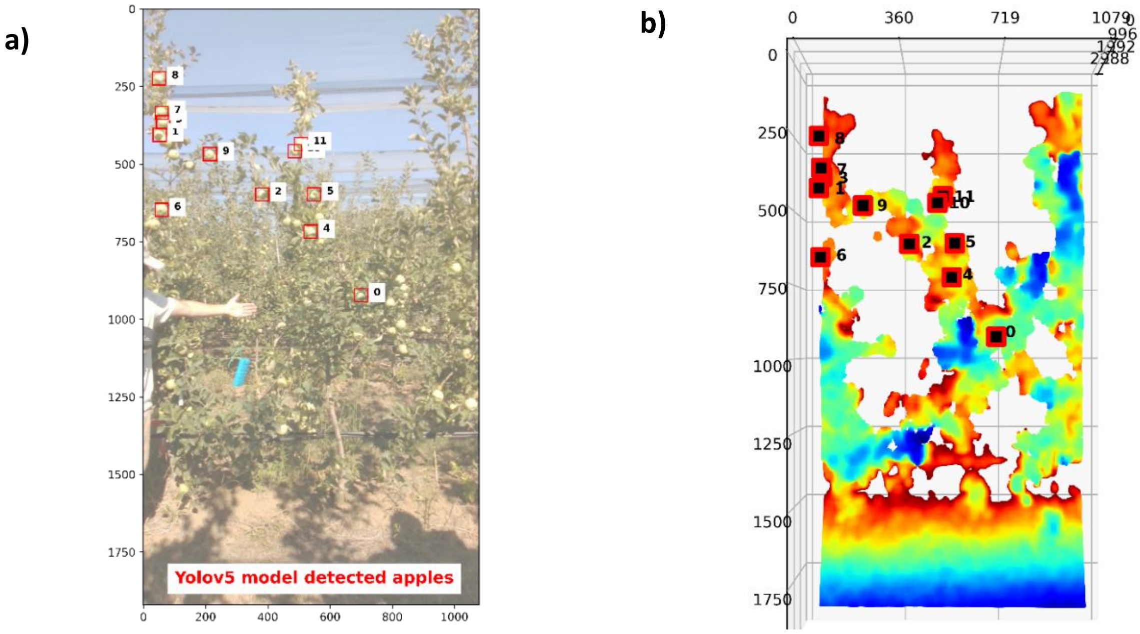

Figure 20. fruit detection projection on the 3D depth map representation of a tree. a) RGB image with fruit detection bbox (red rectangle). b) bbox projection on the 3D represented depth map (i.e, Dbbox – red rectangle).

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 (Figure 14). To account for that, a distance occurrence filtering step is applied to Dbbox: as shown in Figure 21, after the occurrence filtering application, low frequency (< 10-15%) distance are discarded, while others depth data is maintained.

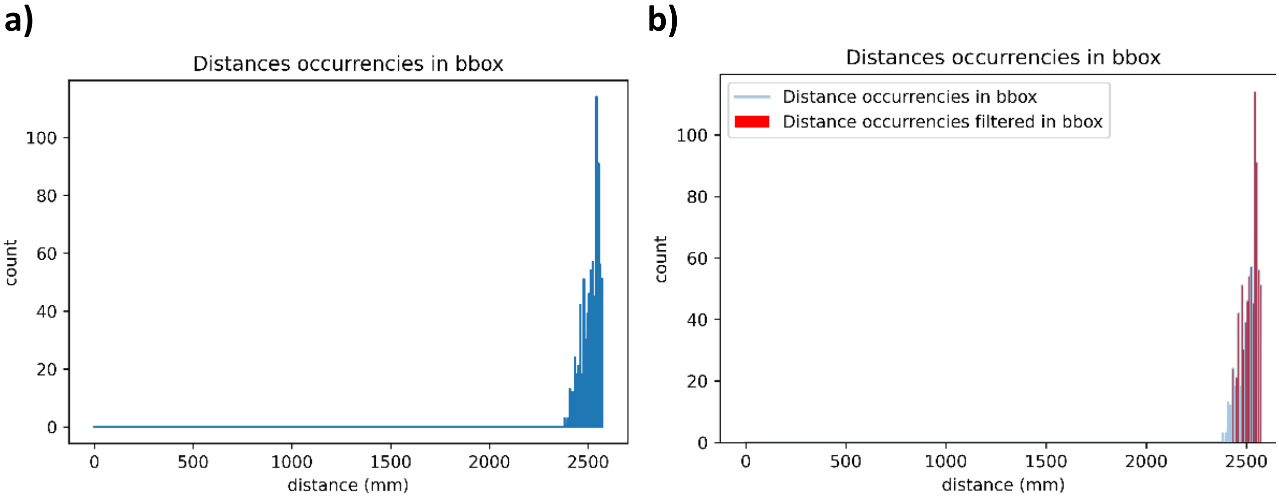

Figure 21. Pre and post occurrence filtering: a) distance occurrences inside the bounding box (blue); b) filtered distance occurrences (red); the highest bar in the plot is considered as the mean fruit distance

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 Figure 22 (a,b).

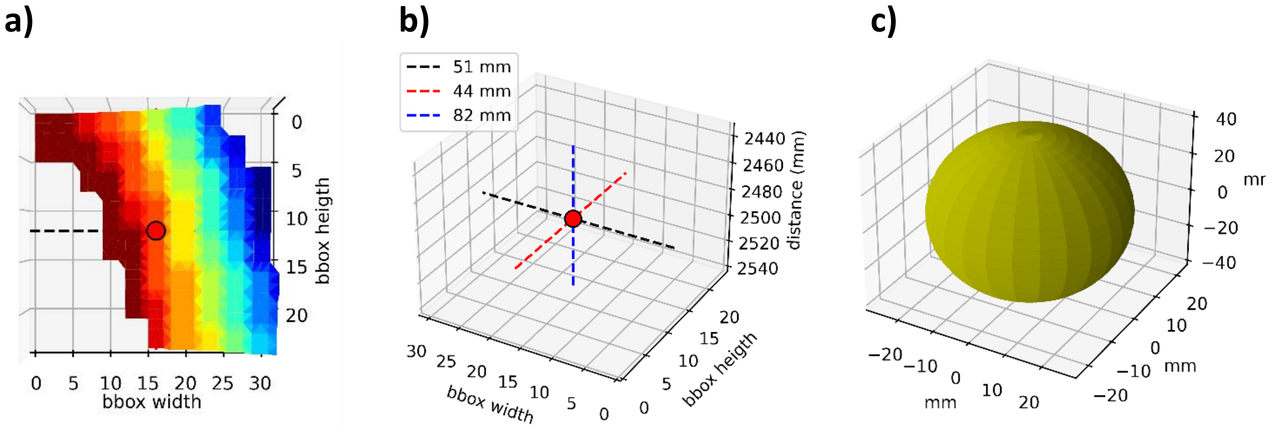

Figure 22.Steps to obtain 3D coordinates of a detected apple: a) filtered depth map with occurrence filtering: the pixel related to the less frequent distances were removed, it can be seen the remaining distances highlighting the convexity of the fruit; b) location of the estimated fruit center at X = X bbox center , Y = Ybbox center , Z= mean fruit distance); c) 3D reconstruction of a possible apple fitting the cubic bbox obtained by bbox widht, 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 Figure 23, Figure 24.

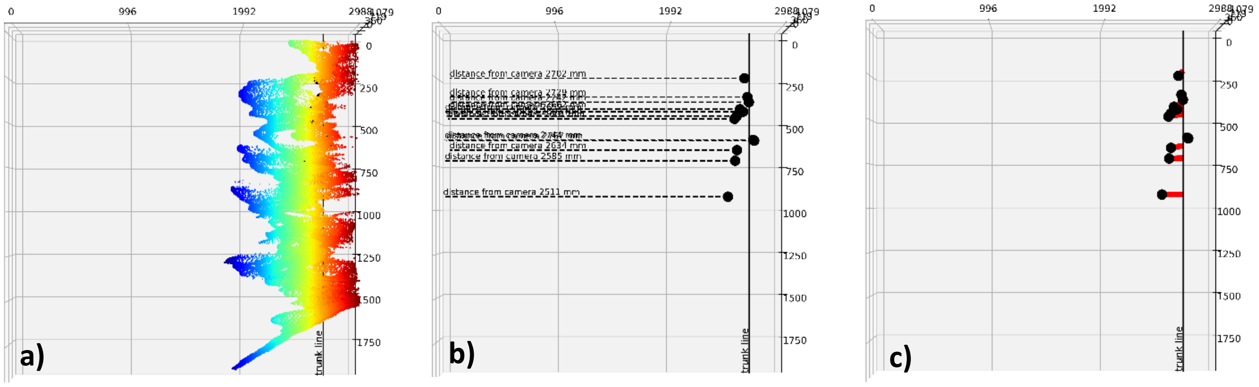

Figure 23. Step to create a 3D positional fruit map of a tree; in the plots, the main represented dimension are Z (horizontal ax) and Y (vertical ax); vertical black line is the trunk origin position. a) the tree filtered depth map; b) the fruit position extraction with coordinates relative to the RGB-D 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 [2], [3].

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 Figure 25.

Evaluation of 3D positional information extraction

The evaluation of the system and process performances in obtaining positional information is currently ongoing. Nevertheless, some preliminary results pointed out that the system seems to present an RMSE of ± 0.1/0.15m 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.
- 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. Figure below show a representation of the results of the whole developed process.

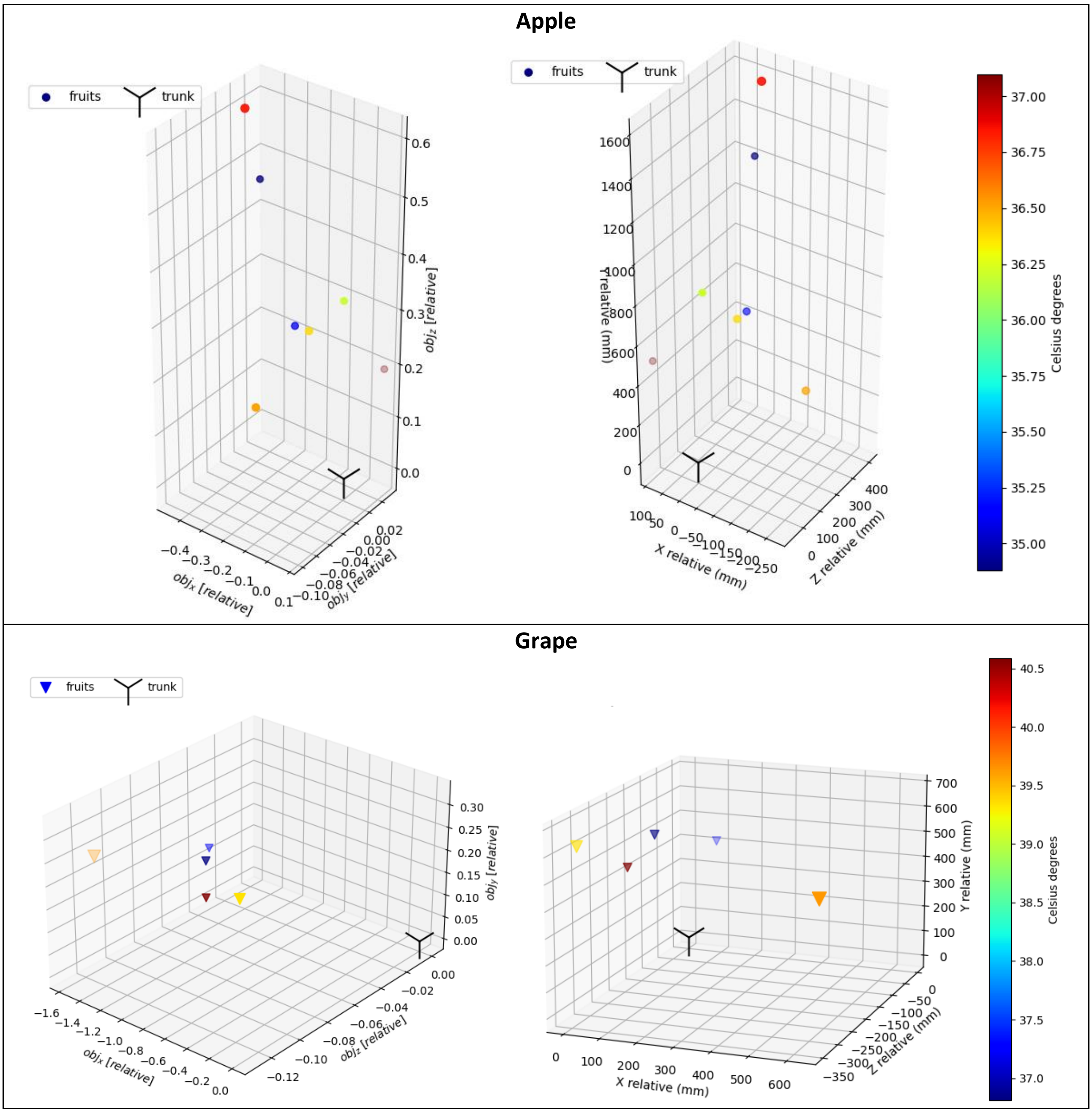

Figure 24. Fruit temperature 3D representation with relative position (pure relative on the left, relative in mm on the right) in respect of the trunk origin. Coordinates are relative to the trunk.

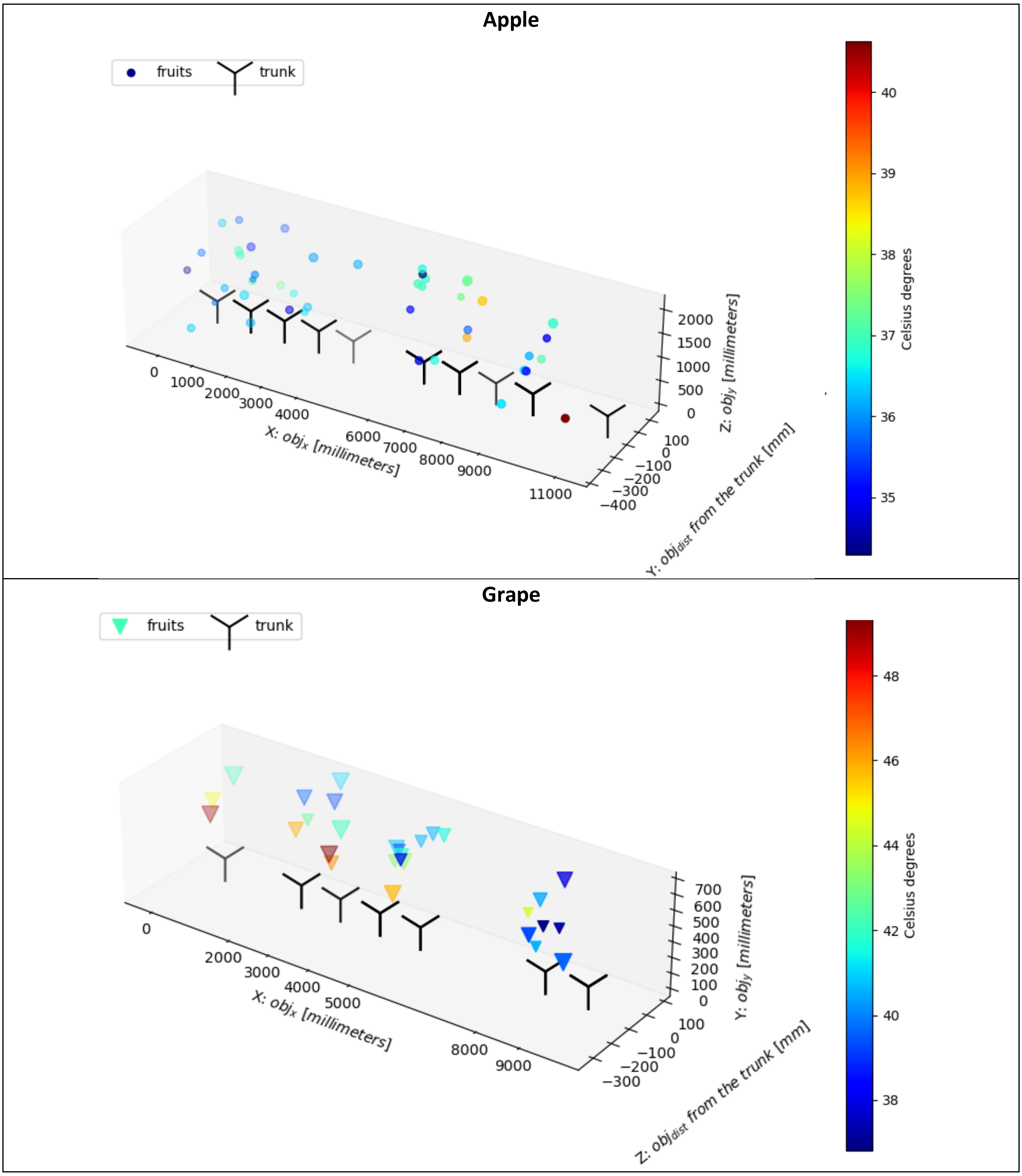

Figure 25. Orchard fruit temperature 3D representation with fruit position relative to a defined orchard origin. Coordinates are in millimeters and relative to the set orchard origin, tree row plane and ground level.

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_alignement_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 →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 on the ATB cloud as “D2.2-2.4-CODE-SHEET_thermal_project-main.zip”.XXXXX

<u>SHEET_project_tree_structure</u>

```
|  ALIGNMENT.py
|  Get_aligment_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
```

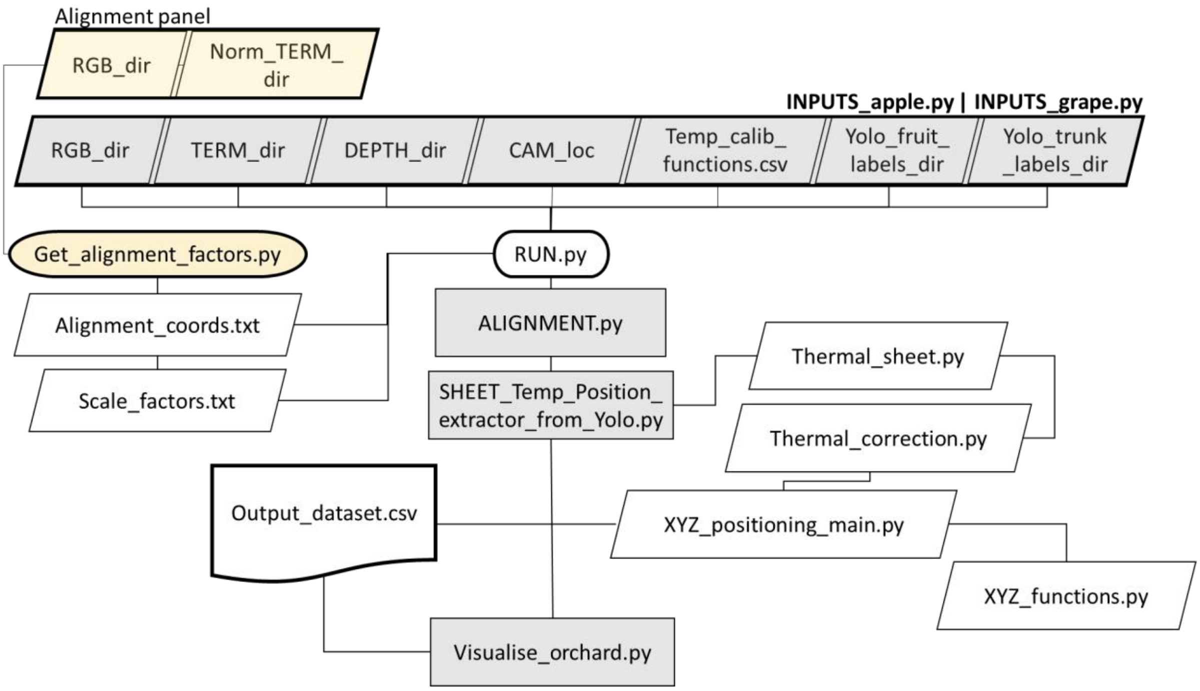

Figure 26. python program workflow