SoftwareX 24 (2023) 101548

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

AKFruitYield: Modular benchmarking and video analysis software for Azure Kinect cameras for fruit size and fruit yield estimation in apple orchards

Juan Carlos Miranda [a,*], Jaume Arnó [a], Jordi Gené-Mola [a,b], Spyros Fountas [c], Eduard Gregorio [a]

[a] Research Group in AgroICT & Precision Agriculture, Department of Agricultural and Forest Sciences and Engineering, Universitat de Lleida (UdL) - Agrotecnio CERCA Center, Rovira Roure 191, Lleida, Catalonia 25198, Spain
[b] Efficient Use of Water in Agriculture Program, Fruitcentre, Parc Científic I Tecnològic Agroalimentari de Gardeny (PCiTAL), Institute of AgriFood Research and Technology (IRTA), Lleida, Catalonia 25003, Spain
[c] Agricultural University of Athens, Iera Odos 75, Athina 118 55, Greece

ARTICLE INFO

Keywords:
RGB-D camera
Fruit detection
Apple fruit sizing
Yield prediction
Allometry

ABSTRACT

AKFruitYield is a modular software that allows orchard data from RGB-D Azure Kinect cameras to be processed for fruit size and fruit yield estimation. Specifically, two modules have been developed: i) AK_SW_BENCHMARKER that makes it possible to apply different sizing algorithms and allometric yield prediction models to manually labelled color and depth tree images; and ii) AK_VIDEO_ANALYSER that analyses videos on which to automatically detect apples, estimate their size and predict yield at the plot or per hectare scale using the appropriate algorithms. Both modules have easy-to-use graphical interfaces and provide reports that can subsequently be used by other analysis tools.

Metadata

Nr	Code metadata description	
C1	Current code version	*1.0*
C2	Permanent link to code/repository used for this code version	https://github.com/GRAP-UdL-AT/ak_sw_benchmarker https://github.com/GRAP-UdL-AT/ak_video_analyser
C3	Permanent link to reproducible capsule	https://pypi.org/project/ak-sw-benchmarker/ https://pypi.org/project/ak-video-analyser/
C4	Legal code license	*MIT license* (http://opensource.org/licenses/MIT)
C5	Code versioning system used	*Github*
C6	Software code languages, tools and services used	*Python 3.8 or later* *Required packages managed with pip: requeriments_win.txt, requirements_linux.txt* *Azure Kinect SDK Pyk4a*
C7	Compilation requirements, operating environments and dependencies	*Windows 10 or Ubuntu Linux 20.04, Azure Kinect SDK, Python 3.8 or later*
C8	If available, link to developer documentation/manual	https://github.com/GRAP-UdL-AT/ak_sw_benchmarker/blob/main/README.md https://github.com/GRAP-UdL-AT/ak_video_analyser/blob/main/README.md
C9	Support email for questions	juancarlos.miranda@udl.cat

(*continued on next column*)

(*continued*)

1. Motivation and significance

In a recent article, Miranda et al. [1] developed AKFruitData, a dual software application available to users to facilitate the use of RGB-D (Azure Kinect) cameras in apple orchard environments. With this initial software, tree data acquisition (videos) was performed to allow a second phase of the creation of datasets for further analysis. Videos were recorded under field conditions containing information about color, depth and IR data from the scene. In the attempt to give continuity to the previous application, this paper focuses on presenting a second software (AKFruitYield) that complements the previous one, now seeking a double research and agronomic objective: i) to allow the benchmarking of different apple fruit sizing algorithms and allometric models with the aim of providing recommendations for the future; and ii) to develop a video analysis tool for RGB-D cameras that, including fruit detection and

* Corresponding author.
E-mail address: juancarlos.miranda@udl.cat (J.C. Miranda).

https://doi.org/10.1016/j.softx.2023.101548
Received 4 August 2023; Received in revised form 27 September 2023; Accepted 1 October 2023
Available online 6 October 2023

sizing, allows for a reliable prediction of fruit yield in apple orchards (per hectare or within a plot).

In order to better organize logistics at the farm level, obtaining reliable early yield estimates is an objective that the use of these new technologies can help to achieve [2–4]. Even efforts have been carried out by publishing software packages in open access to better understand and manage the soil-crop-yield continuum within the framework of Precision Agriculture [5–7]. The use of photonics and computer vision in fruit growing to achieve the aforementioned objective is becoming increasingly popular [3]. The detection, measurement and estimation of fruit size are tasks that can be carried out using different methodologies that include those based on 2D images from those that provide 3D point clouds [8–11]. In the first case, it is necessary the use of calibration targets (known dimensions) that must be located close to the object (fruit) that is aimed to be sized. On the other hand, 3D techniques (e.g., structure-from-motion, LiDAR sensors, RGB-D cameras) generate 3D point cloud reconstructions of the scene. This avoids the use of calibration targets and allows simultaneous estimate the size of all the fruits in the image. Among 3D techniques, RGB-D cameras are a very interesting option given their low cost and the varied and extensive information they offer in each capture [12–15]. Computer techniques together with the use of RGB-D cameras have been applied to different crops [16–22] that are of interest both to the scientific community and industry. However, the pending issue is to have friendly computer applications (automated algorithms) to process the large amount of field data and convert it to useful information for fruit growers and managers. This is the goal of the AKFruitYield software presented in this paper.

The significance of the software that is presented lies in its addressing both fruit size estimation and allometric yield prediction following a comprehensive approach and allowing, at the same time, benchmarking of different algorithms in search of the best combined strategy (detection-sizing-yielding). Unlike certain commercial applications, the fact that information on estimation errors is provided at each stage is a very useful feature, especially for researchers and managers. Briefly, AKFruitYield software can be used as a research tool (increasing knowledge about automatic algorithms in fruit growing) or, from a more applied point of view, as a yield prediction tool covering a need that is increasingly in demand on the part of the fruit sector.

All the data used for the development of this software come from a field test carried out in an experimental apple orchard (cultivar Story® Inored^{cov}) located in Mollerussa, Lleida, Spain. Specifically, trees were arranged according to a 3.6 × 1 m plantation pattern with a canopy height of about 3.5 m (latitude: 41.617465 N; longitude: 0.870730; 246.3 m a.s.l. ETRS89). As mentioned, fruit data was acquired using the Azure Kinect camera (Microsoft, Redmond, WA, USA) on which the development of this software is focused.

2. Software description

The modular structure of AKFruitYield arises from a double requirement: i) to provide a tool for scientific use (AK_SW_BENCHMARKER), in this case a module to design and test different sizing and yield prediction algorithms for apple fruits; and ii) to complement the first with another second more applied tool (AK_VIDEO_ANALYSER) for automatic fruit detection (based on deep learning) on videos recorded with the Azure Kinect camera to then automate fruit sizing and yield prediction algorithms on the detected apples. Both modules (AK_SW_BENCHMARKER and AK_VIDEO_ANALYSER) were developed using the Python 3.8 programming language with Tkinter-based graphic user interfaces (GUIs), making it easier to use the applications on Windows 10 or later and Linux operating systems. In addition, image processing made use of the OpenCV library, with Numpy, Scikitlearn and Pandas being the libraries used for data management. With respect to the implementation of object detectors, PyTorch making use of Mask R-CNN [23] and Faster R-CNN [24] models trained on own apple data from the experimental orchard was the open source learning library used in the video analyser module.

Fig. 1 shows the proposed data acquisition and extraction stages on which the design of AKFruitData and AKFruitYield was based. AKFruitData was designed exclusively for the acquisition and extraction of data from fruit orchards. Once the orchard data is obtained and conveniently organized (Fig. 1a), the AKFruitYield software can be put into operation to, after algorithm training, perform fruit detection, sizing and yield prediction. Interoperability between the specific modules of each software (AKFruitData and AKFruitYield) is shown in Fig. 1b. While the AK_ACQS module is responsible for data acquisition in video format from the orchard (complemented by the AK_SM_RECORDER [25]), the AK_FRAEX module allows extracting RGB images (frames) with additional depth information in different formats [1]. At this point the AKFruitYield software comes in, with the purpose of complementing the functionalities after the acquisition. Using the AK_SW_BENCHMARKER module on extracted frames (PASCAL-VOC format, [26]), different sizing algorithms and allometric models can be combined with final testing of results. In parallel, the AK_VIDEO_ANALYSER module implements the most appropriate sizing and yield prediction algorithms on video records (Matroska MKV format [27]), having previously trained fruit detection deep learning models on frames extracted in COCO format [28].

2.1. AK_SW_BENCHMARKER module

As mentioned, the AK_SW_BENCHMARKER module was designed for algorithm comparison tasks. Using the interface shown in Fig. 2, the software user can select between different size estimation algorithms that provide geometric measurements of the fruit (apple width and height). These size parameters are then used as inputs in different selectable allometric models for the final fruit yield prediction (weight in g).

The AK_SW_BENCHMARKER module offers the user two main tabs with grouped functionalities (Fig. 2): 'Dataset metrics' (Fig. 2a) and 'Metric comparisons' (Fig. 2b). Using the software starts with the selection of the dataset (labelled image in PASCAL-VOC format created by the previous AK_FRAEX module in the AKFruitData software), with the 'Dataset metrics' tab open (Fig. 2a). Once the camera parameters are pre-configured (Azure Kinect in our case), the user must establish the region of interest (ROI) selection method to be applied to the images. Two approaches are available ('ROI selector' in Fig. 2a): assigning bounding boxes to the detected fruits (BBOX method) or delimiting the fruit region by a binary mask (MASK method).

The next phase is to estimate the size in pixels of the perpendicular axes that define the geometry of the fruit (width or caliber and height). In the case of delimiting the ROI by means of the BBOX method, the caliber and height are obtained by adjusting the bounding boxes (BBs) to the labelled fruit. If, instead, a binary mask is used to delimit the ROI, the user can select between four different options ('Size estimation selector' in Fig. 2a) for adjusting geometric figures taking the ROI mask as a reference: circle enclosing (CE), circle fitting (CF), ellipse fitting (EF), and rotated rectangle (RR).

'Depth selector' (Fig. 2a) is the drop-down menu where the distance from the fruit to the RGB-D camera (depth) can be estimated by calculating a metric on the ROI pixels. Available metrics are the average depth (AVG), the modal depth (MOD) or the minimum depth (MIN). This data is then used to convert the geometric measurements of the fruit (pixels) to measurements of width and height in mm (thin lens theory). The final step is to choose an allometric model from those proposed in the drop-down menu below ('Weight prediction method' in Fig. 2a).

The functionalities developed in the AK_SW_BENCHMARKER module (Fig. 2) are listed below. In either case, summary information is presented on screen ('User info' section), and reports are stored as files in an output directory.

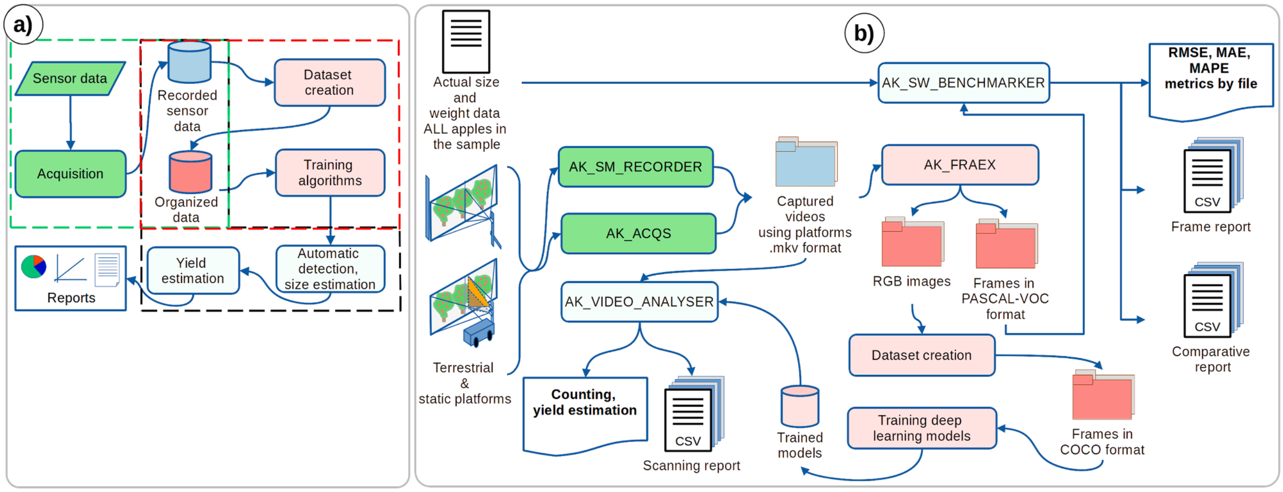


Fig. 1. (a) Proposed stages of data acquisition and extraction for AKFruitData and AKFruitYield. Dashed green lines correspond to processes related to acquisition, red lines to processes related to data creation and training, and black lines to processes for performance estimation. b) Interoperability between the data acquisition (AK_ACQS; AK_SM_RECORDER), data creation (AK_FRAEX), algorithm benchmarking (AK_SW_BENCHMARKER) and video analysis (AK_VIDEO_ANALYSER) modules. The processes proposed in Fig. 1 are expanded and represented by the developed software.

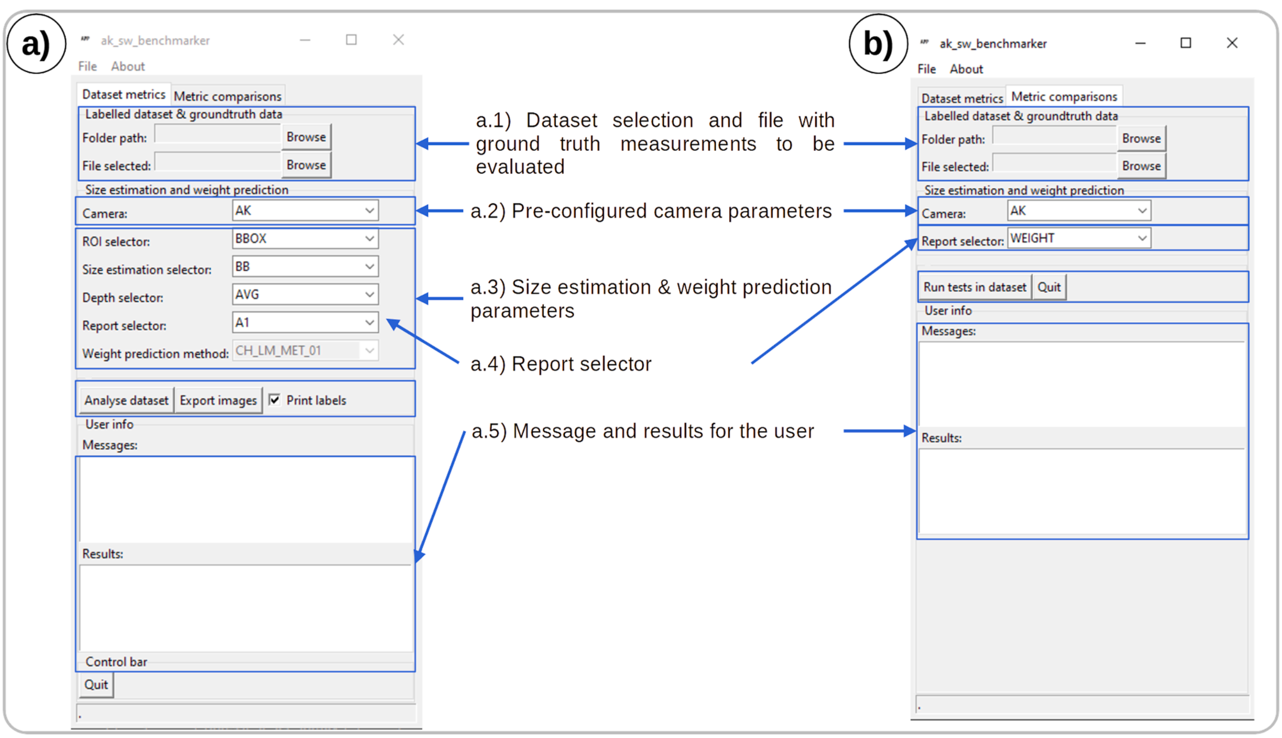


Fig. 2. AK_SW_BENCHMARKER module user interface. a) 'Dataset metrics' tab to select data (frames) and configure the sizing and yield prediction algorithms. b) 'Metric comparisons' tab to report results and error statistics.

- 'Analyse dataset' allows benchmarking to be performed with a final report in CSV format of size estimates and weight prediction. The user has the option of introducing a file with values of real dimensions of fruits (ground truth) to compare with the set of images that it is desired to be analysed. A final report with results (size and weight) grouped by image and fruit will be presented according to the selected parameters in addition to the following error metrics: root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).
- 'Export images' makes it possible to visualize the geometric fitting of the ROIs on the objects (fruits) to be measured. Outputs are color images including binary masks of selected objects and fruit labeling. This functionality adds value to the software since the user can observe how sizing algorithms are applied to the images, enabling corrective adjustments in the algorithm configuration if necessary.
- 'Run tests in dataset' calculates the test metrics. The user must first select the 'Metric comparisons' tab (Fig. 2b) and then, indicate the estimate to be analysed (Report selector in Fig. 2b), namely the major geometric axis of the fruit (A1), the minor axis (A2) or the weight (WEIGHT). Since all sizing-yielding combinations are analysed, this functionality allows the method with least error to be determined, also obtaining a final ranking of sizing algorithms or ranking of sizing-allometric model combinations.

2.2. AK_VIDEO_ANALYSER module

The AK_VIDEO_ANALYSER software module is focused on analysing RGB-D Azure Kinect camera videos. Automatic detection of fruits, sizing of detected fruits and yield prediction (fruit weight) based on allometry are the tasks it performs. Under the same concept as the benchmarking module, a GUI (Fig. 3a) facilitates configuration of the parameters, allowing: i) setting the start and end of the video fragments; ii) filtering by zones and depth; iii) choice of object detector; and iv) choice of sizing and yield prediction methods. In real time, information about counted fruits and yield is displayed to the user as the frames of the video fragments are analysed (Fig. 3b).

Computerized tasks in the AK_VIDEO_ANALYSER module for processing information from video records to final yield results are shown in greater detail in Fig. 4. Now, the RGB-D videos (MVK format) are the starting point for the software. These were previously acquired in orchard environments (in static or from mobile platforms) through the AK_ACQS module that is part of the AKFruitData software [1]. Considering a frame of a set of video fragments, the software module requires certain image filtering settings to be set (‘Depth and coordinates filtering’ in Fig. 3). In this way, the user can filter a certain spatial location within the frame (filtering by coordinates) or a range of depth, discarding those objects that exceed a certain distance from the Azure Kinect camera. By making appropriate settings from the GUI, objects in the depth image can therefore be discarded by distance to obtain a ’thresholded depth image’ (Fig. 4) which, merged with the original color image, provides the filtered RGB image that serves as input to the detector in further processing. ‘Coordinate filtering’ will then be used to delimit the detection zone.

‘Object detection’ uses a trained model (Mask R-CNN or Faster R-CNN) that is applied to the depth-filtered RGB image (Fig. 4). As objects (fruits) are being detected, size determination (fruit axes) using the ROI data (BBOX or MASK, depending on the method) serves to then enter the geometric measurements of the apples in the allometric model. Reports in CSV format are activated at the end of the analysis. Each fruit (apple) is matched to the measurements of the major axis and the minor axis in pixels, the estimated values in mm, and the predicted weight in g.

The functionalities developed in the AK_VIDEO_ANALYSER module (Fig. 3) are listed below.

- ’Analyse video’ allows the user to configure video analysis parameters. Examples are the number of frames to analyze, filters to apply, or detection, sizing and weight prediction models to be implemented. Results are displayed on screen and conveniently organized in a CSV file.
- ’Preview video’ helps to configure detection zone dimensions and distance filters on color images.
- ’Export frames’ provides the user with a set of analysed images and the information obtained. It is a useful functionality to observe how algorithms are applied on the frames.
- ’Reset settings’ allows the user default values in the GUI to be reset.
- ’Run in command line’ allows video analysis using the command line without the need for a GUI screen. Useful functionality in carrying out scriptable processes.

3. Illustrative examples

An example of sizing algorithm benchmarking for final yield prediction is shown in Fig. 5. First (Fig. 5a), selection of the appropriate options in the GUI of the AK_SW_BENCHMARKER module causes binary masking to achieve ROIs of labelled apples to be bounded, to then fit ellipses (EF) to the ROIs over the color image. A detail of the operation for the apple with label 2136 is shown in Fig. 5b. Red points on the edge of the ROI serve as a guide for fitting the ellipse (in green) and obtaining the semi-major and semi-minor axes of the apple marked in blue and green, respectively.

Extracts in Fig. 5c work as follows. Ground truth data of size and weight per labelled apple (c.1) are shown together with the intermediate results (prediction data, c.2). Final report (c.3) includes the compared ground truth and predicted data of geometric length (major axis 01 or minor 02) or weight at the request of the software user. The last piece of information in Fig. 5d shows an overview report. In the case of the figure, the user can view the results of the error metrics which, by assessing the major axis (A1), have been obtained according to different sizing and depth algorithms on the set of labelled apples for which real size data are available. In this way, the user has comparative information between methods (algorithms). It is worth mentioning that preliminary tests with the AK_SW_BENCHMARKER have provided fruit size estimates (non-occluded fruits) with MAPE < 5 %. These results values are comparable to those obtained with other state-of-the-art 3D sensing

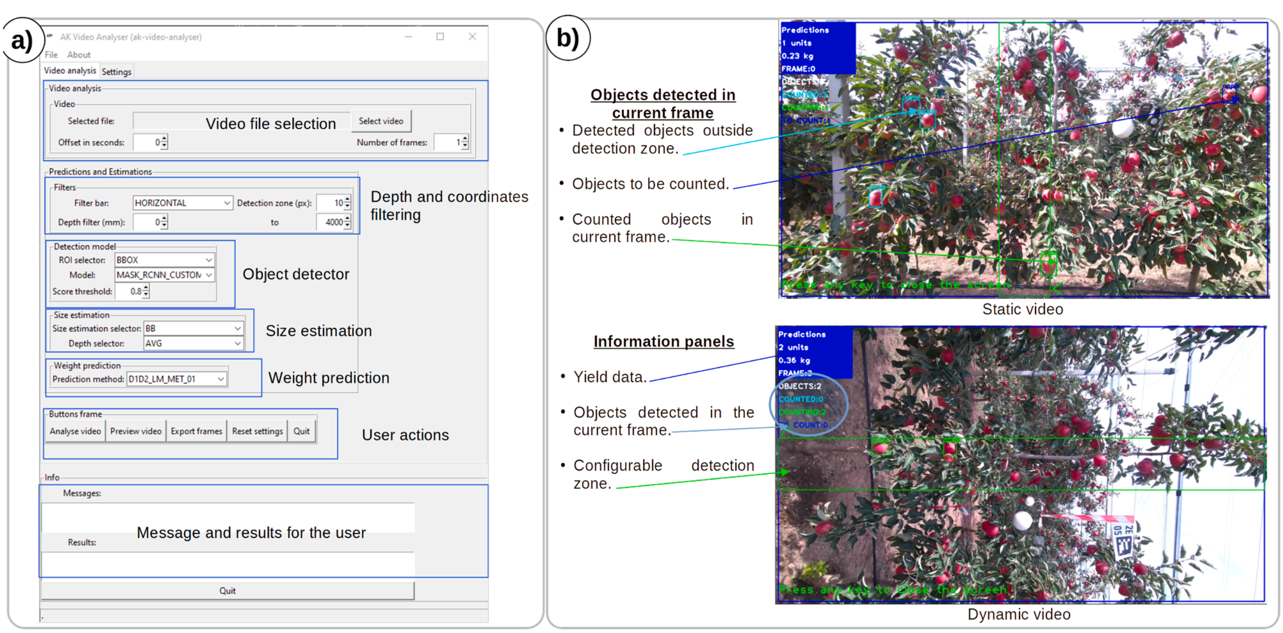


Fig. 3. AK_VIDEO_ANALYSER module user interface. a) Main GUI. b) Output screen showing detected fruits and report of results in real time.

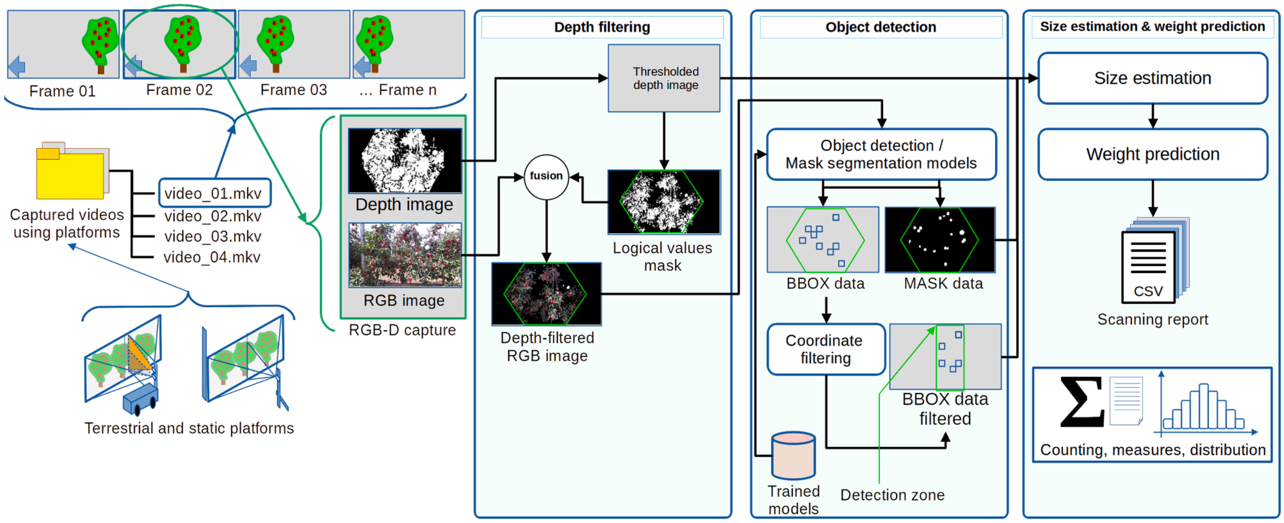


Fig. 4. Tasks that are performed sequentially by the AK_VIDEO_ANALYSER module that is part of the AKFruitYield dual software.

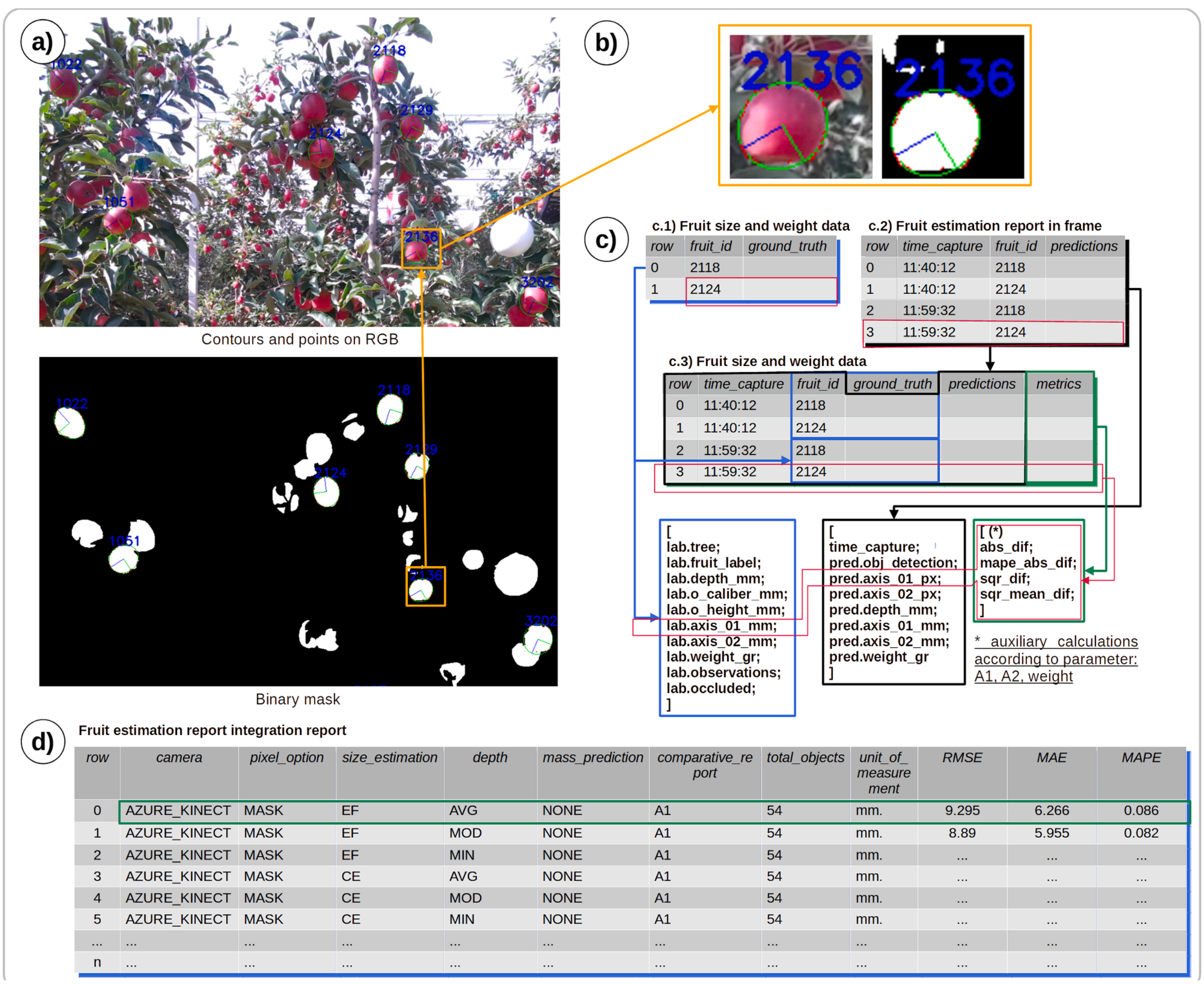
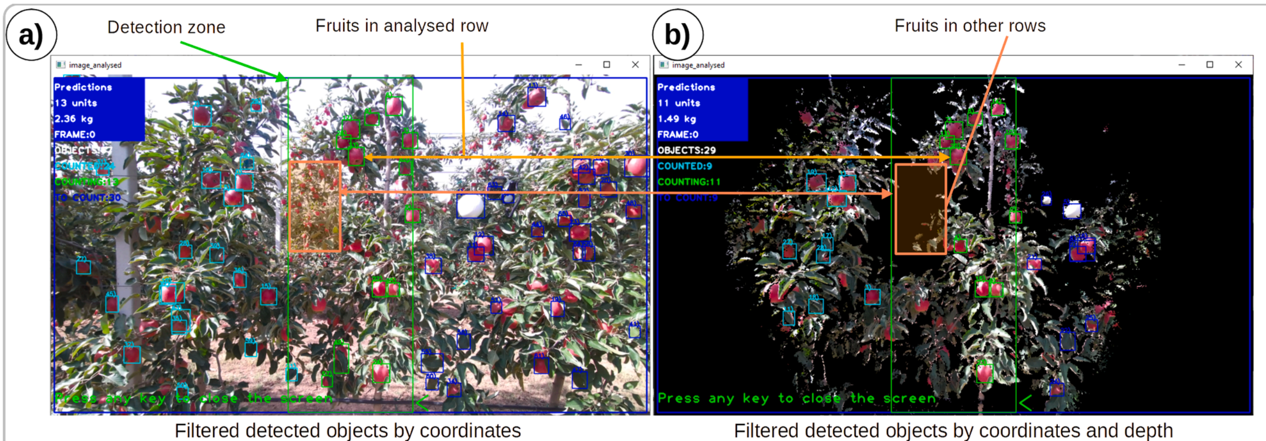


Fig. 5. Example of the operation of the functionalities of the AK_SW_BENCHMARKER module.

techniques [9]. The AK_SW_BENCHMARKER have allowed to identify the best combinations of sizing algorithms and allometric models for which fruit weight predictions with a MAPE < 6 % were achieved, lower than the 10 % of error threshold usually accepted in yield predictions.

As for the other software, the AK_VIDEO_ANALYSER module, some of its features can be seen in Fig. 6. Use of the module is initially argued



c) **Fruit estimation prediction report**

row	frame	pred.obj_detection	fruit_id	pred.axis_01_px	pred.axis_02_px	pred.depth_mm	pred.axis_01_mm	pred.axis_02_mm	pred.weight_gr
0	0	0	1_0	64	61	1193.03	73.417	69.976	174.929
1	0	1	1_1	42	36	1347.708	54.427	46.651	87.573
2	0	2	1_2	64	55	1227.202	75.52	64.9	184.603
3	1	3	1_3	55	55	1366.466	72.265	72.265	169.629
4	1	...	...	...	...	...	...	...	...
5	1	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
n	...	...	...	...	...	...	...	...	...

Fig. 6. Example of some functionalities of the AK_VIDEO_ANALYSER module.

for analysis of videos recorded with the Azure Kinect camera on trees within an apple orchard. However, the module has the possibility of being adapted and used in other orchards if detection algorithms and weight prediction models are made available.

Capture of three apple trees in a given row are shown (Fig. 6a). Note that apples belonging to trees in the back row are detected (orange box) by coordinates and distance filtering (Fig. 6b) and are not counted. This is a very useful functionality to delimit the fruits to be detected and processed, avoiding errors in final yield predictions at a spatial plot scale. As the AK-VIDEO_ANALYSER module works, three detection zones are shown to the user corresponding to: i) fruits already counted (in light blue); ii) fruits being counted and real use phase of sizing and weight algorithms (in light green); and iii) detected fruits that will be analysed shortly (in dark blue).

Real-time informative data is displayed in the upper left-hand part of the screen. At the end of the analysis, a report (information shown in Fig. 6c) summarizes size (in pixels and mm), distance location (depth in mm) and weight (in g) for each detected fruit. In addition to these tests, algorithms have been applied to foam spheres (visible in Fig. 6). Consequently, different examples of apples and foam spheres are included in the software repository to check the applicability of the AK_VIDEO_ANALYSER software module on two different types of objects.

4. Impact

With the development of the AKFruitYield software (AK_SW_-BENCHMARKER and AK_VIDEO_ANALYSER modules), functionalities of other related and open access software (AKFruitData [1]) are complemented. Two tools are therefore made available to the user to allow the use of RGB-D Azure Kinect cameras to, in a first step, acquire data in orchards and create analysable datasets (AKFruitData) and, in a second step, to process the data (videos and images) and provide reliable fruit yield predictions (AKFruitYield).

The impact of AKFruitYield (in combination with AKFruitData) is expected to be important for two main reasons: i) by providing fruit growers and, especially, managers in fruit growing an application that can significantly reduce the time and cost of a task still carried out manually in many farms; and ii) by promoting the introduction of low-cost accessible technologies under a strategic framework of digitizing the fruit sector.

AKFruitYield is open source software designed to allow future updates. New detectors can be implemented, sizing algorithms improved and allometric models refined as its use adds more knowledge and users demand new specifications (other apple tree cultivars, including other fruit species, or applications in earlier stages of fruit development as a scouting tool). For the present, AKFruitYield provides a practical solution in terms of yield prediction (and not simple fruit counting) in apple orchards. Additionally, a powerful benchmarking module is made available to test other detection, sizing and yield allometry options that may be implemented as a result of its use. Multi-Object Tracking and Segmentation (MOTS) [29] is another powerful option for greater precision that could be implemented in the future, once testing and the required adaptations were made in the current version of the software.

AKFruitYield (presented here) and AKFruitData [1] have been designed to encourage the use of the Azure Kinect camera (and depth cameras in general) by end users working or doing research in fruit growing. Ease of use is a key aspect and, for this reason, video examples and image datasets to verify the operation of the software are attached as supplementary material to the source code. It is possible to use this software with RGB-D sensors other than the Azure Kinect camera. However, implementing specific data extraction routines (wrappers) according to the selected devices would be required.

5. Conclusions

AKFruitYield is presented in a modular format since it includes two separate but complementary modules focused on processing fruit data acquired through RGB-D Azure Kinect cameras in apple orchards. The first module (AK_SW_BENCHMARKER) makes it possible to apply different sizing algorithms and allometric yield prediction models on color and depth tree images containing previously detected and manually labelled apples. The second module (AK_VIDEO_ANALYSER) is a video analysis software that allows automatic apple detection, size estimation and yield prediction to be performed by applying the best ranked algorithms resulting from the previous benchmarking. A special objective has been to develop easy-to-use graphical interfaces for end users working in research as well as fruit growers and advisory technicians who require new digital tools for better management of fruit farms.

Future works are planned to expand the current functionalities and provide support under the concept of continuous development and improvement. In short, with the AKFruitYield software, the cycle that started with the AKFruitData software is successfully completed, thus meeting the initial objectives of acquiring and processing data from Azure Kinect cameras in apple orchards for yield prediction.

Declaration of Competing Interest

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

Data availability

The links to the research data/code are [https://github.com/GRAP-UdL-AT/ak_video_analyser/] and [https://github.com/GRAP-UdL-AT/ak_simulator/].

Acknowledgments

This work was partly funded by the Department of Research and Universities of the Generalitat de Catalunya (grants 2017 SGR 646) and by the Spanish Ministry of Science and Innovation/AEI/10.13039/501100011033/ERDF (grant RTI2018–094222-B-I00 [PAgFRUIT project] and PID2021–126648OB-I00 [PAgPROTECT project]). The Secretariat of Universities and Research of the Department of Business and Knowledge of the Generalitat de Catalunya and European Social Fund (ESF) are also thanked for financing Juan Carlos Miranda's pre-doctoral fellowship (2020 FI_B 00586). The work of Jordi Gené-Mola was supported by the Spanish Ministry of Universities through a Margarita Salas postdoctoral grant funded by the European Union - NextGenerationEU. The authors would also like to thank the Institut de Recerca i Tecnologia Agroalimentàries (IRTA) for allowing the use of their experimental fields, and in particular Dr. Luís Asín and Dr. Jaume Lordán who have contributed to the success of this work.

References

[1] Miranda JC, Gené-Mola J, Arnó J, Gregorio E. AKFruitData: a dual software application for Azure Kinect cameras to acquire and extract informative data in yield tests performed in fruit orchard environments. SoftwareX 2022;20:101231. https://doi.org/10.1016/j.softx.2022.101231.
[2] Bargoti S, Underwood JP. Image segmentation for fruit detection and yield estimation in apple orchards. J Field Robot 2017;34(6):1039–60. https://doi.org/10.1002/rob.21699.
[3] He L, Fang W, Zhao G, Wu Z, Fu L, Li R, Majeed Y, Dhupia J. Fruit yield prediction and estimation in orchards: a state-of-the-art comprehensive review for both direct and indirect methods. Comput Electron Agric 2022;195(February):106812. https://doi.org/10.1016/j.compag.2022.106812.
[4] Zhang C, Valente J, Kooistra L, Guo L, Wang W. Orchard management with small unmanned aerial vehicles: a survey of sensing and analysis approaches. Precis Agric 2021;22(6):2007–52. https://doi.org/10.1007/s11119-021-09813-y.
[5] Conti G, Mercante E, de Souza EG, Sobjak R, Bazzi CL. AGDATABOX-RS computational application: remote sensing data management. SoftwareX 2023;23:101435. https://doi.org/10.1016/j.softx.2023.101435.
[6] Correndo AA, Pearce A, Bolster CH, Spargo JT, Osmond D, Ciampitti IA. The soiltestcorr R package: an accessible framework for reproducible correlation analysis of crop yield and soil test data. SoftwareX 2023;21:101275. https://doi.org/10.1016/j.softx.2022.101275.
[7] Bazzi CL, Jasse EP, Graziano Magalhães PS, Michelon GK, de Souza EG, Schenatto K, Sobjak R. AgDataBox API – Integration of data and software in precision agriculture. SoftwareX 2019;10:100327. https://doi.org/10.1016/j.softx.2019.100327.
[8] MacEachern CB, Esau TJ, Schumann AW, Hennessy PJ, Zaman QU. Detection of fruit maturity stage and yield estimation in wild blueberry using deep learning convolutional neural networks. Smart Agric Technol 2023;3:100099. https://doi.org/10.1016/j.atech.2022.100099.
[9] Miranda JC, Gené-Mola J, Zude-Sasse M, Tsoulias N, Escolà A, Arnó J, Rosell-Polo JR, Sanz-Cortiella R, Martínez-Casasnovas JA, Gregorio E. Fruit sizing using AI: a review of methods and challenges. Postharvest Biol Technol 2023;206:112587. https://doi.org/10.1016/j.postharvbio.2023.112587.
[10] Parr B, Legg M, Alam F. Grape yield estimation with a smartphone's colour and depth cameras using machine learning and computer vision techniques. Comput Electron Agric 2023;213:108174. https://doi.org/10.1016/j.compag.2023.108174.
[11] Gregorio E, Llorens J, Kerry R, Escolà A. Sensing Crop Geometry and Structure. Sensing approaches for precision agriculture. progress in precision agriculture. Cham: Springer; 2021. https://doi.org/10.1007/978-3-030-78431-7_3.
[12] Condotta ICFS, Brown-Brandl TM, Pitla SK, Stinn JP, Silva-Miranda KO. Evaluation of low-cost depth cameras for agricultural applications. Comput Electron Agric 2020;173:105394. https://doi.org/10.1016/j.compag.2020.105394.
[13] Pasinetti S, Nuzzi C, Luchetti A, Zanetti M, Lancini M, De Cecco M. Experimental procedure for the metrological characterization of time-of-flight cameras for human body 3D measurements. Sensors 2023;23(1):538. https://doi.org/10.3390/s23010538.
[14] Gené-Mola J, Llorens J, Rosell-Polo JR, et al. Assessing the performance of RGB-D sensors for 3D Fruit crop canopy characterization under different operating and lighting conditions. Sensors 2020;20(24):7072. https://doi.org/10.3390/s20247072.
[15] Lopes A, Souza R, Pedrini H. A survey on RGB-D datasets. Comput Vis Image Underst 2022;222:103489. https://doi.org/10.1016/j.cviu.2022.103489.
[16] Hacking C, Poona N, Manzan N, Poblete-Echeverría C. Investigating 2-D and 3-D proximal remote sensing techniques for vineyard yield estimation. Sensors 2019;19(17):3652. https://doi.org/10.3390/s19173652.
[17] Wang W, Li C. Size estimation of sweet onions using consumer-grade RGB-depth sensor. J Food Eng 2014;142:153–62. https://doi.org/10.1016/j.jfoodeng.2014.06.019.
[18] Wang Z, Walsh KB, Verma B. On-tree mango fruit size estimation using RGB-D images. Sensors 2017;17(12):1–15. https://doi.org/10.3390/s17122738 (Switzerland).
[19] Sun M, Xu L, Luo R, Lu Y, Jia W. Fast location and recognition of green apple based on RGB-D image. Front Plant Sci 2022;13. https://doi.org/10.3389/fpls.2022.864458.
[20] Liu J, Yuan Y, Zhou Y, Zhu X, Syed T. Experiments and analysis of close-shot identification of on-branch citrus fruit with RealSense. Sensors 2018;18(5):1510. https://doi.org/10.3390/s18051510.
[21] Xiong Y, Peng C, Grimstad L, From PJ. Isler V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. Comput Electron Agric 2019;157(January):392–402. https://doi.org/10.1016/j.compag.2019.01.009.
[22] Gené-Mola J, Vilaplana V, Rosell-Polo JR, Morros JR, Ruiz-Hidalgo J, Gregorio E. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. Comput Electron Agric 2019;162:689–98. https://doi.org/10.1016/j.compag.2019.05.016.
[23] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE; 2017. p. 2980–8. https://doi.org/10.1109/ICCV.2017.322.
[24] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2017;39(6):1137–49. https://doi.org/10.1109/TPAMI.2016.2577031.
[25] Miranda J.C., Gené-Mola J., Arnó J., Gregorio E.. AK_SM_RECORDER, https://pypi.org/project/ak-sm-recorder/, 2022 [accessed 26 September 2023].
[26] Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vis 2010;88(2):303–38. https://doi.org/10.1007/s11263-009-0275-4.
[27] The Matroska Association. Matroska, https://www.matroska.org/, [accessed 26 September 2023].
[28] Lin TY, Maire M, Belongie S, Fleet D, Pajdla T, Schiele B, Tuytelaars T, et al. Microsoft COCO: common objects in context. Computer Vision – ECCV 2014. Springer International Publishing; 2014. p. 740–55. https://doi.org/10.1007/978-3-319-10602-1_48.
[29] de Jong S, Baja H, Tamminga K, Valente J. Apple MOTS: detection, segmentation and tracking of homogeneous objects using MOTS. IEEE Robot Autom Lett 2022;7:11418–25. https://doi.org/10.1109/LRA.2022.3199026.