Information management and monitoring system for a grapes harvesting robot

K Tziridis, A Nikolaou, T Kalampokas, E Vrochidou, T Pachidis, G A Papakostas and V G Kaburlasos

Human-Machines Interaction Laboratory (HUMAIN-Lab), Department of Computer Science, International Hellenic University, Agios Loukas, 65404 Kavala, Greece

E-mail: gpapak@teiemt.gr

Abstract. In recent years, there has been an increasing integration of high technology in the execution of agricultural tasks towards improving the quality of the produced products, while reducing production costs. In this context, there is a special interest in the development of automatic agricultural robots (Agrobots) with advanced techniques of collecting, processing data, decision making and applying actions. A key element of the successful integration of such autonomous robots is the information management and monitoring system (IM2S) that accompanies it. This work presents the IM2S developed for an autonomous grapes harvesting robot, as part of a national project. Details regarding the system’s design are presented, covering system requirements, functional specifications, Graphical User Interface (GUI) design, database design and system-robot communication. Along with the presentation of the IM2S design, the special functional and technological requirements that must be met, are also discussed. Thus, the system could be a useful tool for the human operator. Moreover, the designed system can be easily adapted to collaborate with any robot meeting specific communication requirements. This makes the proposed IM2S a universal tool for any grape harvesting Agrobot already deploy or will be developed in the future.

1. Introduction
The technological advancement of automation and robotics, consequently resulted to the advent of robots in agriculture, able to drastically increase the productivity, reduce the operating costs and the lead time of agriculture [1]. Agricultural robots, namely Agrobots, are currently used towards the automation of agricultural tasks. Recently, several research attempts propose vineyard agrobots for different purposes; pruning [2], watering [3], harvesting [4], spraying [5]. The new generation of agrobots demonstrates a higher level of sophistication using artificial intelligence for decision-making on the spot. The agrobot of our interest is an autonomous grape harvesting robot (AGR), as part of a co-funded national project. ARG is developed to perform harvest, green harvest and defoliation, based on commercial hardware, machine vision and innovative computational intelligence algorithms. The developed algorithms are intended for real-time processing of the sensory data, and decision making under the robot’s sensory awareness [6,7]. The robot consists of a mobile vehicle, a robotic arm with adequate end-effector and mounted sensors such as light detection and ranging (LiDAR), a global positioning system (GPS) and cameras.

The focus of the proposed work is on information management and visual monitoring for ARG. The motivation behind the proposed work lies in reducing human labor and bringing efficiency in dexterous grape-related agricultural operations through the use of an autonomous agrobot. The novel
contributions of this work are: 1) a solid information management system that can be generalized and adopted by different projects that meet the same requirements, 2) a vineyard monitoring system based on inexpensive sensors and 3) a friendly interactive software, specially developed to monitor the vineyard addressed to people with minimum knowledge of technology.

The rest of the paper is organized as follows. System requirements are presented in section 2 and the functional specifications in section 3. Database design and GUI are shown in sections 4 and 5, respectively. Section 6 summarizes the system-robot communication. Finally, section 7 concludes the paper.

2. System requirements

The IM2S application that communicates and interacts with the agrobot is an application designed to run on a portable computer. Thus, the central unit of the system will be a laptop, placed in the operating range of the robot, in order to send and receive real-time information to/from the robot. The system requirements for the hosting machine so as to run smoothly and interact loosely with the robot is a Windows 10 64bit operating system with at least 1.5GB hard disk space for the application installation and 4GB disk space for the vineyard maps. The orthomosaic vineyard maps are extracted from the acquired images of an unmanned aerial vehicle (UAV) by using the Agisoft software. The application uses the vineyard maps to navigate the robot between the vineyard rows [8]. Additionally, it should have at least 8GB RAM memory and a CPU of at least i5-2540m. There are no GPU requirements for the laptop, due to the fact that the application’s main purpose is to preview and organize the received information from the robot and to send back commands to the robot through the database. Moreover, the robot is supported by Robot Operating System (ROS), uses MongoDB database and needs a WiFi connection.

3. Functional specifications

The users need to Login and have access to vineyard maps of their interest. All actions of the users need to be stored to an individual history file. The administrator needs to register new users and update the application. The administrator can access all maps and history files. The user can see all real-time measurements of the robot and live stream from its cameras, knowing exactly from which part of the vineyard these videos come from. All measurements and video sequences need to be able to get saved in files. Measurements need to grouped and visualized on the maps in an understandable way. Measurements regarding the status of the robot, such as battery level, the current position of the robot, working time, also need to be displayed. The users need to be able to select the agricultural task for the robot and customize the selected task from a setting panel based on their personal practices. The IM2S needs to keep a history of the tasks and choices made by the users. Statistics regarding the selected tasks, such as the number of bunches removed, how long the work took to complete, etc., should also be recorded. Since the measurements are numerous, the users need to have the possibility to choose which of the information will be displayed at the screen at any time. It should be possible for the users to check that all the individual components that comprise the robot work properly and are connected to each other. The users need to be able to alternate between their maps and see all possible paths on every stored vineyard map. Moreover, the users need to be able to select one or more paths on each map, where the robot should navigate and perform the selected task. If the users do not select a path or specific settings for an agricultural task, then the system will use default settings and perform the tasks on the closest path from its current location. However, the users need to select the agricultural task to further proceed, so that the system to send their options to the robot and start executing the task. The users need to be able to zoom in/out on the displayed maps so as to clearly inspect the paths and measurements shown on the maps. The users need to be able to give the command to the robot to start a task but also to stop working at any time.

4. Database design

The database of the IM2S is the most important part of the application since it binds the communication of the application with the robot. It is a MongoDB non-relational database and it is selected due to its ability to store JavaScript Object Notation (JSON) documents instead of data in
tables, like SQL. This architecture permits the storing of different data categories to different collections. In figure 1 it is illustrated a part of the database diagram which shows extensively the type of data exchanged between the database and the application, and between the database and the robot.

![Database diagram.](image)

5. Graphical User Interface (GUI) design

The User collection stores all the information about the accounts that have access to the application. More analytically, the data of this collection are primarily the usernames and passwords of the users. Additionally, the history of every user is stored along with the vineyard maps of users’ interest. The Maps collection stores the vineyard maps that are provided to the user. In these maps, the crop rows are identified and the navigation route mapping of the robot is designed, i.e. the paths between the crop rows where the robot should navigate and execute the agricultural tasks. In this collection, the geographic coordinates of the paths (the start and end points) are also stored. The Spatial maps collection stores 3 different maps. These maps are formulated on the main vine map, based on the collected data of the robot during its navigation in the vineyard paths. The three spatial maps are depicting temperature, moisture and vegetation. Thus, the user can have a complete image of the aforementioned metrics distribution in the vine, in real-time. The Path Planning collection is created based on the orthomosaic maps and it stores the path coordinates of the maps that the user selects. More specifically, the user selects the exact paths on the vineyard map of his interest, where the robot should navigate to and perform agricultural operations. The Robot Info collection stores information about the robot parts (manipulator, end-effector, encoders etc.) so as the user to be able to monitor the system’s hardware and to detect in time any malfunction, knowing from which part it came from. Moreover, this collection includes information regarding the robot’s speed, battery level, the traveled distance and its current longitude and latitude. The Metrics collection includes measurements of humidity, temperature, chlorophyll, grapes ripeness level, number of grape clusters and number of leaves that the robot removed. The Job Parameters collection stores the settings for the three agricultural tasks; harvest, green harvest and defoliation. These settings concern the leaves and grapes to be removed and are defined by the user. Finally, the Image Stream collection stores the images from the robot’s cameras. All the information is stored and displayed at the GUI for real-time monitoring.

The GUI of the application is implemented with Qt5. This minimal user interface is meant to be as simple as possible, addressed to the average user. The GUI displays all relevant information regarding the agricultural tasks that the robot can execute and regarding the robot itself. The GUI comprises the Login Option window and the Main GUI window which consists of 7 tabs.

In the Login Option window, one can select to login either as a user or as an administrator. In the user login window, the user needs to enter his account credentials, i.e. username and password that are stored in the database. If the inserted credentials exist in the database, then the user can continue to the
Main GUI window. In the administrator login window, the same process takes place for the administrator to enter the GUI. When entered, the administrator can create accounts for new users. The administrator creates the credentials for the new users, which are stored in the database at the User collection, giving them access to the GUI.

The Main GUI window consists of the following seven tabs:

- **Maps**: this is the main tab. The user can load the desired orthomosaic map, can choose the paths on the map where the robot will navigate to and can choose the agricultural operation that the robot will perform. The Start button is inactive until the Job parameters, which are found in the next tab, are defined. Then, the Start button becomes available for sending all the parameters to the robot. The user can press at any time the Stop button to interrupt the robot’s operation.

- **Job parameters**: the user defines the parameters of the each agriculture task. For the defoliation, the user can choose the number of leaves to be removed and from which side of the vineyard (east or west). For the green harvest, the user can choose how many grape bunches to remove per tree and the way to remove them; starting from both edges or alternately per shoot. For the harvest, there are no parameters to be defined. The robot removes and collects all remaining grape bunches on the vines. After the user had selected the task parameters and had stored them, for the application is ready to send them to the robot. All information from this tab is stored in the Job Parameters collection.

- **Connections**: the user can monitor the robot components’ status. With a green indicator next to them are marked the components that work properly while with a red indicator those in which a problem is detected. The information about this tab are retrieved from the Robot Info collection of the database.

- **Monitor**: the user can access useful information about the vineyard, the selected robot task and the robot components. This information is retrieved from the Metrics collection, the Robot Info collection and the Image Stream collection of the database. The actions that a user can do in this window are to hide a selected group of the information that is not of his interest, to save all displayed information in a .csv file and to save images from the stream.

- **Spatial maps**: the user can see Spatial maps collection information displayed on the orthomosaic map of the vineyard. Three different maps can be displayed and alternate. The selected metric to be displayed is presented as a dot on the orthomosaic map, located at the exact point on the map where the robot took the corresponding measurement. All spatial maps can also be stored by the user.

- **History**: the user has access to all previous actions. A list of all up-to-date selected tasks, settings that have been sent to the robot along with timestamps, is provided. The user can export this data in a JSON file and preview it at any time.

- **Help**: at this tab the user can find the manual of the GUI.

The main tab (Maps) of the GUI, is illustrated in figure 2, and the “Spatial maps” tab in figure 3.

6. System-Robot communication

The robot while functioning, it senses the environment and gathers data. This data has to be sent back to the main computer, to the system’s database in real-time to update the IM2S and the displayed information on the GUI for correct decision making by the user. On the other hand, the user needs to send data about the tasks and settings to the robot, stored through the GUI, in the database and transmitted to the robot so as to start or/and stop executing. Therefore, the robot and the system write and read data to and from the database. In order to set up a real-time communication, between the robot and the main computer system, a Wi-Fi based communication network is established. Therefore, the used mobile robot vehicle needs to have Wi-Fi communication capabilities. Otherwise, the mobile robot needs to be controlled via a Wi-Fi wireless network with the help of simple software and hardware, such as chip-based modules that are well-known cost-effective tools of performing Wi-Fi wireless control functions.
Figure 4 illustrates the communication between the application and the robot. As it can be seen, the IM2S—robot communication depends on Wi-Fi. The database runs on the host computer, where the desktop application is installed, in order to avoid overloading the robot’s internal disk drive. The application writes a json file into the database, containing details regarding the desired working area for the robot and all parameters of the agricultural task that will be performed. When the robot receives all details, it starts executing. While operating, it collects data from the environment which sends back to the database. Simultaneously, the collected data are shown in real-time on the monitor window of the application.

7. Conclusions
The application of agricultural robots in viticulture has experienced an increase in investment and research. In this work, an information management and monitoring system for an autonomous grapes harvesting agrobot is presented. System requirements, functional specifications, GUI design, database
design and system-robot communication are some of the aspects that are discussed. The proposed system is meant to be a useful tool for the robot’s operators. Moreover, the designed system can be easily adapted to collaborate with any robot that meets the communication requirements. This is due to the fact that the system depends on a robotic platform running on ROS. All robots that run on top of ROS come with their vendor packages preinstalled. Thus, the proposed IM2S is generic and can be considered as a universal tool that can be generalized for any grape harvesting agrobot.

Figure 4. IM2S-robot communication.

References
[1] Mavridou E, Vrochidou E, Papakostas G A, Pachidis T and Kaburlasos V G 2019 Machine vision systems in precision agriculture for crop farming, *J. Imaging* 5 89
[2] Botterill T, Paulin S, Green R, Williams S, Lin J, Saxton V, Mills S, Chen X and Corbett-Davies S 2017 A Robot System for Pruning Grape Vines, *J. F. Robot*, 34 1100–22
[3] Thayer T C, Vougioukas S, Goldberg K and Carpin S 2018 Routing algorithms for robot assisted precision irrigation, *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, pp. 2221-2228, doi: 10.1109/ICRA.2018.8461242
[4] Luo L, Tang Y, Lu Q, Chen X, Zhang P and Zou X 2018 A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard, *Comput. Ind.*, 99, pp. 130–9
[5] Adamides G, Katsanos C, Constantinou I, Christou G, Xenos M, Hadzilacos T and Edan Y 2017 Design and development of a semi-autonomous agricultural vineyard sprayer: Human-robot interaction aspects, *J. F. Robot*, 34, pp 1407–26
[6] Kalampokas T, Tziridis K, Nicolaou A, Vrochidou E, Papakostas G A, Pachidis T and Kaburlasos V G 2020 Semantic Segmentation of Vineyard Images Using Convolutional Neural Networks, *21st International Conference on Engineering Applications of Neural Networks (EANN 2020)*, pp 292–303
[7] Badeka E, Kalabokas T, Tziridis K, Nicolaou A, Vrochidou E, Mavridou E, Papakostas G A and Pachidis T 2019 Grapes Visual Segmentation for Harvesting Robots Using Local Texture Descriptors, *12th International Conference on Computer Vision Systems (ICVS 2019) (Thessaloniki)*, pp 98–109
[8] Badeka E, Vrochidou E, Tziridis K, Nicolaou A, Papakostas G A, Pachidis T and Kaburlasos V G 2020 Navigation Route Mapping for Harvesting Robots in Vineyards Using UAV-based Remote Sensing, *IEEE 10th International Conference on Intelligent Systems (IS)*, pp 171–7

Acknowledgments
This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (T1EDK-00300).