Multi-Agent Collaborative Framework for Automated Agriculture

Kumar Ankit[1], Shishir Kolathaya[2] and Debasish Ghose[3]

***Abstract*— The use of internet-connected devices, especially small multi-rotor Unmanned Aerial Vehicles (UAVs), in scientific data gathering and applications, is quite widespread. But due to limited intervention capability, the UAVs alone fail to automate agricultural tasks completely. Thereby, we propose a centralized framework capable of handling a heterogeneous mixture of UAVs and UGVs to cater to the needs of automating agriculture efficiently. The framework's core is a novel heuristic decision module that creates new tasks by visually analyzing the farm and solves a vehicle routing problem to allocate it to agents optimally. It is also equipped with supporting modules to monitor their operation and, in case of failures, help them recover autonomously based on the task and agent assessment. The framework is used in three significant agricultural applications, namely yield prediction and drought stress detection in a simulated environment using ROS and Gazebo, and 3D mapping of a real farm. These applications demonstrate the use of the multi-agent collaborative framework in identifying agricultural tasks in a farm and executing them.**

***Index Terms*— ulti-UAS collaboration, Vision based pick and place, Task planning, Fault Handlingulti-UAS collaboration, Vision based pick and place, Task planning, Fault HandlingM**

I. INTRODUCTION

The world's population in 2015 stood at 7.3 billion people and is expected to increase to 8.5 billion by 2030. It requires a sustainable and efficient farming methods to meet the food demands with constantly decreasing farmlands. Sustainable farming also known as precision agriculture has been in limelight in academia since long back but lacks focus on a complete autonomous solution. With expansion of internet-connected devices, new ways to measure and analyze agricultural processes in real-time came into the picture. Internet of Things (IoT) in agriculture can be considered a viable solution as it needs constant monitoring and intervention. Using IoT in agriculture needs appropriate software architecture that plays a prominent role in optimizing the gain. In the case of IoT, a drone can act as a mobile data collector and transfer it to a remote station for processing. In practice also an aerial vehicle is generally deployed for data collection and surveillance whereas most of the heavy duty farm operations are carried out by UGVs. Due to the complexities involved with autonomous agriculture, an aerial or ground vehicle alone is not sufficient to carry out agricultural tasks. Hence a common autonomous platforms is necessary that can serve as a framework in which multiple heterogeneous IoT devices and systems can be incorporated and controlled to carry out tasks jointly. In this work, we propose a heterogeneous multi-agent framework employed in an agricultural setup with UAVs and UGVs to carry out precision agriculture tasks autonomously.

*This work was supported by PMRF Program of Govt. of India

[1]Kumar Ankit is a doctoral student at Robert Bosch Center for Cyber Physical Systems (RBCCPS), Indian Institute of Science, Bangalore, India kumarankit@iisc.ac.in

[2]Shishir Kolathaya is in the Department of Computer Science and Automation and RBCCPS, Indian Institute of Science, Bangalore, India shishirk@iisc.ac.in

[3]Debasish Ghose is in the Department of Aerospace Engineering and RBCCPS, Indian Institute of Science, Bangalore, India dghose@iisc.ac.in

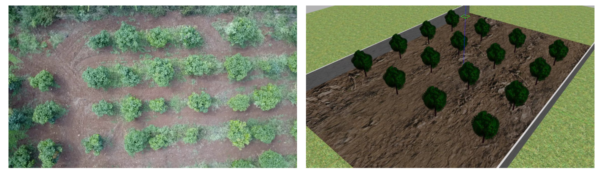

Fig. 1. a) Original Farm (Left) b) Simulated Farm (Right)

UAVs and UGVs are being deployed for a diverse range of applications in precision agriculture. Reduction in waste and chemical emissions increased overall efficiency, and profitability is some of the best-achieved results. In [1], the authors use a quadrotor with an on-board RGB camera for crop monitoring, health assessment, and spraying. Software architecture built to promote reuse, support farms of various sizes is proposed in [2]. A framework named SmartFarm [3] which integrates environmental sensors present in farm with a private cloud infrastructure that provides farmers with a secure, easy-to-use, and cost-effective data analysis system. Similarly, an autonomous multi-sensor UAV system is proposed for remote sensing, which is adequate for small fields [4]. An architecture for collective field monitoring by a group of UAVs is realized in [5]. A swarm of UAVs is deployed in [6] for spraying fertilizers over agricultural land. Yield prediction of coffee crops using images from UAVs is also made in [7]. Indices such as the Normalized difference vegetation index, which depicts the produce's health, are derived from the UAV's multispectral imagery [8]. We see that The heterogeneous use of UAVs and UGVs is a promising combination to be applied that has been utilized as mobile robot UGVs are used to serve as a transport and recharge station for the UAVs in [9].

Collaborative mapping of construction sites [10] using both UAV and UGV is also being done. The use of a heterogeneous swarm of edge UAVs for remote sensing is used in [11]. Infrastructure-free cooperative relative localization in GPS-denied environments [12] can also be realized when collaborating. A novel collaborative 3D map registration pipeline is presented in [13] for agricultural applications.

The framework proposed in this paper uses a master-slave architecture with the master as a base station and agents as slaves. The framework's core is a heuristic decision module that creates new tasks, decides, and optimally allocates it to agents. The tasks are the series of actions that need to be performed by the agents over a particular region that we define as action zones. Action zones can be a pest infected area, which requires the spraying of pesticides. The framework is composed of modules such as path planner, task scheduler, etc., and it can be easily integrated with specific modules depending upon the tasks. This allows the framework to be applied to most primary agriculture operations, from sowing seed to harvesting the produce. The framework has been deployed in a simulated agricultural environment based on the actual farm layout, as shown in Fig. 1.

We discuss our proposed framework in Section 2. Section 3 provides an experimental evaluation to highlight the efficacy of our approach and value due to simplicity. In Section 4, we consider two of the use-cases and simulated the same in Section 5. Finally, we conclude the paper in section 6 with remarks and discussion on future work.

II. Collaborative Multi-Agent Framework

In this section, we describe the framework and its functional modules:

A. *Objective of the Framework*

Given *n* UAVs and *m* UGVs, the aim is to detect action zones and deploy these *n+m* agents to perform their desired functions in those action zones. The UAVs are equipped with a camera to capture images. The UGV carries a 6-DOF manipulator with a sprayer and camera at the end-effector. Periodically, one of the *n* UAVs is sent to capture complete farm's images in an exhaustive flight path (for example, in a lawnmower pattern). It streams the captured images back to the central base station, which in our case is the master. The base station then processes the imagery to detect action zones and plans the agents' paths according to the zone's requirements and capabilities, creating a route plan. This generated plan is sent to the task-scheduler module, which then schedules the tasks and allocates them to all the agents. The agent then performs the task and creates a report about the progress and status of the vehicle. The report is sent to the master to decide and schedule the next task for the agent. Furthermore, this framework is also designed to handle minor faults such as communication loss or task failure. In such cases, a fault handler module is invoked on both master and slave to help recover from failure. The architecture and modules are described in detail in the section below.

B. *Components of the Framework*

1) Slave Architecture: The proposed architecture (see Fig. 2) with the necessary modules for agents or slave in our case is described below:

i) Communication Module: It is responsible for monitoring communication between the master and the slave. If the connection is broken, the module calls the fault handler for further help to restore the connection.

ii) Agent Handler: It monitors the status of the agent. It gathers information from various on-board sensors and generates the status which includes position, battery health and resources on-board the agent. A tabular description of agent report is shown in Table 1. The sub-modules associated with the agent handler are:
   a) Localization Module: It fuses the inertial and global measurements to get an estimate of the pose and orientation of the agent.
   b) Vision Module: It handles the image stream captured by the camera sensor and streamed back to the master for further processing to detect the action zones.
   c) Status/Health Monitoring: It checks whether the sensors and actuators are operational. It also monitors the on-board resources such as battery-health and fertilizer volume.

TABLE I

Agent Report Description

Particular	Information	Data Type
Pose	$[x, y, z]$	[3x1] Array
Orientation	[roll, pitch, yaw]	[3x1] Array
Battery Health	Battery Left (%)	Integer
Actuator Health	Healthy or not	Bool
Sensor Health	Healthy or not	Bool
Resource Status	Quantity Left	Float

iii) Task Handler: It is responsible for handling the tasks sent to the agents. It receives the task and a location depicting the action zone. Then it uses one of the below-mentioned task-modules to deploy the agent on the task. After which it reports an assessment back to the master. A tabular description of task report is shown in Table 2. The sub-modules associated with the task handler are:
   a) Record: It helps the agent record or map a particular bounded region on the farm. It records the geo-tagged and timestamped images at the height closest to the crops so as to have the highest resolution. The recorded imagery is then streamed back to the master for further analysis.
   b) Travel: The UAV can traverse from point to point at a particular height in air whereas, the UGV has to travel across the rows in grid-wise manner. This sub-module gets the location of the action zones and helps the agent traverse the environment while avoiding obstacles and forbidden zones.
   c) Spray: It is designed to precisely spray a controlled volume of water or chemical at a particular area using the on-board camera.

TABLE II

Task Report Description

Task Report		
Particular	Information	Data Type
Task Code	Record/Travel/Spray	[3x1] Array
Task Position	$[x, y, z]$	[3x1] Array
Task Progress	Percentage(%)	Integer

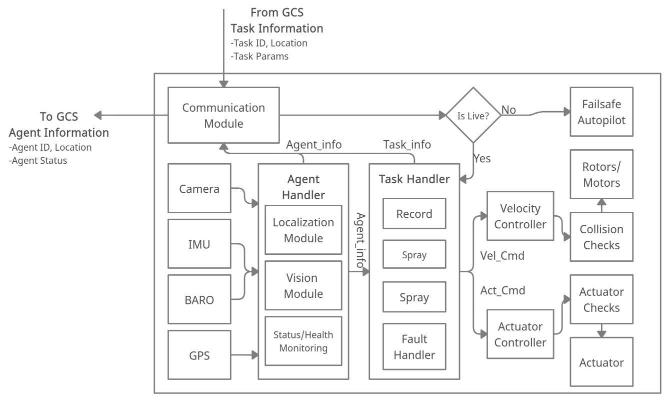

Fig. 2. Slave architecture and its modules

2) Master Architecture: The proposed architecture (see Fig. 3) with the necessary modules for base station or master is described below:

i) Communication Module: It operates in the same manner in both the master and the slave architecture.

ii) Agent Handler: It handles the information received from the agent about its health and status. Based on the report received, it decided whether the agent is operational or not. If operational, the agent is fed with the next task using the task-scheduler module, or else the fault handler module is invoked.

iii) Task Handler: It handles the task plan and assesses whether the assigned task is complete or has failed to complete. If it has failed, the module invokes the decision-maker module to decide the next course of action, or else the agent is allotted the next task using the task-scheduler.

iv) Decision Maker: This is at the core of the master decision-making process. This module decides what needs to be done next with the reports received from the task and agent handler modules. If all the agents are free or no new task remains to be allotted, then the module decides to create new tasks using the task generator module. Once the action zones are detected, or new tasks have been created, the module allocates them to the free agents optimally using the route planner sub-module, creating a route-map. The route-map is then sent to the task-scheduler, which schedules the next task to the agents as soon as a particular agent's last task is complete. If any of the operational agents fail to complete a task, the decision module is invoked to decide whether to retry or skip the failed task. This decision is made based upon how much progress was made on the same.

v) Task Generator: This module is triggered when there is no new task to be carried out by the agent or the agents have accomplished the tasks. For instance, a systematic exploration in a lawn-mover fashion of the farm is carried out by a UAV to detect the action zones and create new tasks. The master instructs a UAV with a given demarcated area to capture close-up imagery at a particular height and send it back to the master. The imagery sent is then processed by the master using the vision module and outputs the stress zones or measurement zones. This information is then sent to the route planner to allocate the agents with the practical tasks according to their capability/capacity.

vi) Vision Module: Responsible for mapping, detecting action zones and converting the pixel-based location into a real-world co-ordinate to be handled by the route planner. The module is designed for the two use cases and simulated (See Section IV). The present framework allows the vision module to be used for the following tasks.

a) Yield Prediction: Imagery captured from UAV contains a tree's view from four different directions and the top to cover it completely. The aim is to count the fruits (such as apples) on top of the green leafy background, as seen in the captured images (see Fig. 5a). The algorithm uses hue, saturation, and value in the HSV color space as visual cues for red apple detection. If the hue value of a particular pixel lies in the red region, then the pixel is marked for segmentation. The segmented object is counted if its area in the image is also greater than some threshold to reject some false detections. The count is then used to estimate the final yield of the farm.

b) Drought Stress Detection: For this, the farm images are captured from the top view at a particular height using UAVs. The images contain leafy ground with some light-colored patches to depict stressed areas (see Fig. 5b). The images are divided into three respective color channels. To detect the stressed areas, gradient change is monitored in the green channel using a Sobel filter. Pixel is considered a boundary if the gradient change lies within a region bounded by a lower and upper threshold. Upper and lower thresholds

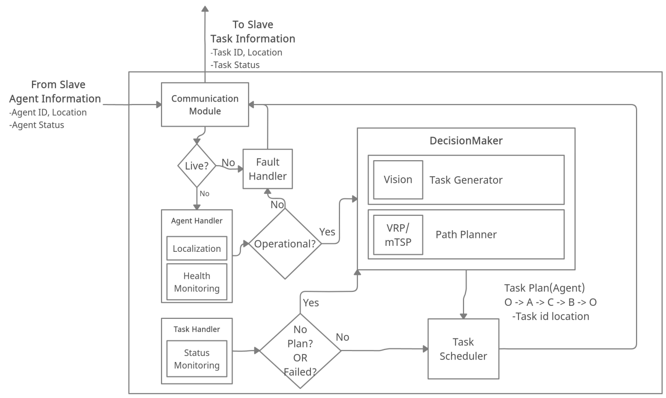

Fig. 3. Master architecture and its modules

are chosen to detect the patch boundary and reject any other gradient change in the image.

c) Collaborative Mapping: Like drought stress detection, the images of a particular area have been captured from the top view at a specific height using UAVs and the crop row using UGV. Agents are sent periodically with the target waypoints to record geo-tagged imagery. The module then employs a novel offline mapping pipeline named AgriColMap [11], suitable for solving structure from motion problems in the agricultural domain. It takes the captured images as the input and outputs the 3D point cloud. The 3D point cloud is then annotated with the detected parameters of the crop. Finally, the point cloud can be used by the decision-maker to detect action zones and plan the agents' routes accordingly.

vii) Route Planner: Given the location of the zones and agents, the module plans the route using Clarke & Wright savings algorithm (see algorithm-1) as our task allocation problem boils down to a multi-vehicle routing problem. Clarke & Wright's savings algorithm [12] plans the route for the agents based upon the requirements of the action zone and the capability/capacity of the agents. For the simulation, we have assumed requirements to be identical at all action zones and all the agents equally capable. Generated route plan is then sent to the task scheduler to handle. An example of a route plan with four agents (3 UAVs and UGV) and 48 action zones depicting trees on the grid has been planned using the algorithm and shown in Fig. 4.

viii) Task Scheduler: The module allocates the task to the agents sequentially according to the route-plan. If the previous task has been completed successfully, a new task is assigned, or else the decision-maker module is invoked in case of any partial or incomplete task.

ix) Fault Handler: This module is invoked when the communication with the agent is lost or the agent has failed. The module is designed to locate and reconnect/retrieve the agent. The module is in an elementary state where the failure is restored by homing the agent and rebooting it. When rebooted, the master tries to set up the communication link and sends the last task it failed to complete. In case of hardware failure, the agent is removed from the operational list, and the routes are planned again for the remaining UAVs and incomplete tasks.

III. Application of The Framework

In agriculture, from sowing seeds to harvesting fruits, the significant chunk of tasks can be divided into sub-tasks as event detection, traveling, and action or treatment. For instance, automated spraying can be done by detecting an infected patch of the crop, traveling to the location, and then treating it with pesticides.

At present, only a recording and a spraying module have been developed for the proof of concept of the framework. The actions can relate to anything given that the agent knows how to perform it. In other words, it has appropriate hardware and software for the task. For deploying agents with the right task, a central base station is brought into play to allocate and assign tasks to the agents. The central base station can be mobile (UGV) or stationary.

In the following sections, we describe the different critical scenarios significant in the agricultural domain [13], where our framework can be deployed.

A. Yield Prediction

The accuracy of the first crop forecast is essential for farmers and the entire agricultural sector. Data for better yield estimates were taken from UAVs using RGB-based crop height and canopy construction or multispectral imagery. Yield prediction becomes necessary in those experimental contexts where nitrogen, phosphorus, or irrigation-related treatment causes more significant variation on the final yield. UAV yield prediction studies to date have focused on the development of energy efficiency models. We have primarily focussed upon the accuracy of the prediction. Hence, more data

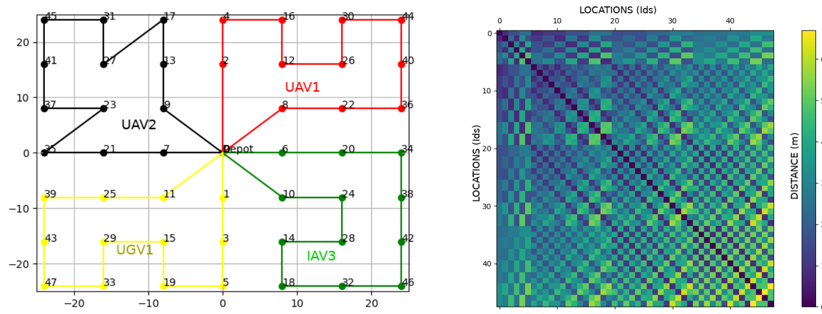

Fig. 4. a) Route Plan for the agents for 48 locations (Left) b) Distance Matrix depicting distance between locations i and j (Right)

Algorithm 1 Clarke & Wright Savings Algorithm

i) Compute the savings

$$s(i, j) = d(D, i) + d(D, j) - d(i, j) \tag{1}$$

for every pair (i, j) of demand points where d(i, j) is the distance between point i and j and D denotes depot or starting point.

ii) Rank the savings s(i, j) in descending order of magnitude, creating the "savings list." Start with the topmost entry in the list (the largest s(i, j)).

iii) For the savings s(i, j) under consideration, include the link (i, j) in a route if no route constraints will be violated through the inclusion of (i, j) in a route, and if any one of them applies:

   A) Neither i nor j has already been assigned to a route. In that case, a new route is created, joining both i and j.

   B) Exactly one of the two points (i or j) has already been included in an existing route. The point is not interior to that route (a point is within the route if it is not close to depot D in the sequence of points). In that case, the route (i, j) is added to that same route.

   C) Both i and j have already been included in two different existing routes, and none of the points is interior to its route. In that case, the two routes are joined.

iv) If the savings list s(i, j) is completely traversed, stop: the solution consists of the joined routes using step 3. otherwise, return to Step 3 and process the next entry.

points are recorded to estimate the yield. Although, a UGV with its close imaging capability can significantly enhance the prediction accuracy [17]. This calls for a collaborative framework to deal with the above.
For this purpose, a patch of land (can be whole agricultural land) is selected by demarcating the vertices with the GPS co-ordinates. The decision-maker is then used to plan the path for the agents using the path planner module. Once the route plan is generated, the task is to reach the targets and record the images to cover the whole patch. This imagery is then sent to the master to detect and quantify the produce.
Counting or detecting produce on-farm is prone to occlusion and false detection. Hence it is used as a marker of the density of the produce. For this purpose, we have assumed three levels of density as low, moderate, and high. Initially, this density classification is based on simple heuristics on the count. But as we get more information after harvesting, this heuristics is updated. This explains how many high, medium and low density trees are there on the field. The farmer can use this information to plan the distribution of fertilizer and irrigation, perform variable crop thinning, and improve operations by increasing efficiency, reducing inputs, and increasing yield over time in underperforming sections.

B. Drought Stress Detection

The sustainable and optimal use of water resources through precision irrigation techniques constitutes the most critical challenge in the agricultural sector. Precision irrigation bears importance for combating water scarcity and curb salinity and loss of nutrients. The method is also integral in preventing the lower-level regions from water clogging and ensuring sufficient water supply in higher-level areas.
This is generally done using thermal imaging cameras as water transpiration directly impacts the surface temperature and color. An RGB camera on UAV can detect the discoloration, followed by the site inspection by UGV with a thermal camera on-board. Studies suggest using the crop water stress index (CWSI) to determine the extent of stress [15] and the amount of water to be sprayed.
In our case, To detect the drought stress zone, the UAV is sent in a lawnmower pattern to record the images from the top view. The geo-tagged imagery is then sent to the base station to process and detect the zones by segmenting them based on the ground's color. The difference in the shade is then used to calculate the quantity of water required. The patch is bounded by a

contour whose center is chosen to be the target location. To generate the location, a pixel to GPS conversion is done, and finally, the action zone is created. The action zone is then allotted to an agent to reach, detect the contour and spray the water over the patch based on the requirement.

C. Collaborative Mapping

Mapping in an agricultural domain done is a routine manner recording the images of the farm periodically. The imagery is then processed offline to generate a 3D point cloud for the same. For this, a structure from a motion library named AgriColMap [11] can be used, which takes images (both aerial and ground imagery) as input and outputs the 3D point cloud solving a large displacement optical flow problem. The 3D point cloud thus generated can be annotated with crop-related information such as crop density and weed pressure, necessary for decision making.
Our framework provides the necessary unified environment for sharing information across the agents and master for such a collaborative library to work smoothly. The decision-maker can also use this 3D point cloud to plan the route for UGV more efficiently. There can be different types of obstacles tackled in an agricultural setup, namely soft and hard obstacles. The soft obstacles can be breached by the agents, whereas the hard ones have to be avoided. For instance, the UGV might consider a weed patch (soft obstacle) as a crop (hard obstacle) while traversing the rows. In that case, we can use the annotated 3D map to detect the weed patch and plan the path according to a soft obstacle.

IV. Simulation Setup and Results

The collaborative mapping module is tested in a real farm whereas, yield prediction and drought stress detection have been tested in a simulated environment due to the farm's unavailability during the pandemic. The simulations are visualized using Gazebo 9.11 and ROS Melodic with PX4 Autopilot for UAV. The following environment is set up to simulate an apple orchard with 3 UAVs and one UGV.

A. Farm

The simulated farm is of the size of 25m x 25m with evenly spaced fruit trees in a grid fashion. The spacing or the row-width considered is 1.5m sufficient for the UGV to travel across. Two sides of the farm are as shown in Fig. 5.

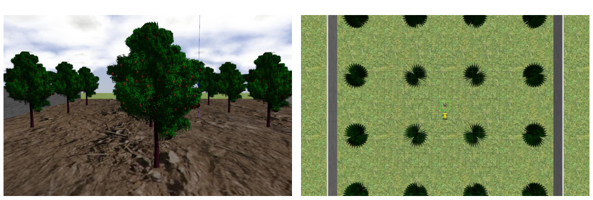

Fig. 5. a) Simulated Apple Tree (Left) b) Simulated Farm (Right)

B. UAVs and UGVs

.

The UAVs (see Fig. 6a) use the IRIS drone model. It is a quadrotor and is equipped with two cameras, one facing downwards for an aerial view and one on the front for close-up imagery acquisition.
For simulating the UGV (see Fig. 6b), the model of Husky by Clearpathrobotics is ported into the Gazebo environment. The UGV is integrated with a UR5 manipulator with a sprayer as an end effector. The UR5 has a reach radius of 850 mm and a payload of up to 5 kg. Two monocular cameras are employed in the UGV, one forward-facing and another downward facing. The forward-facing one aids in search and exploration, while the downward-facing camera on the manipulator is used for orienting the manipulator for spraying operations. The UGV and UAV models are shown in Fig. 6.

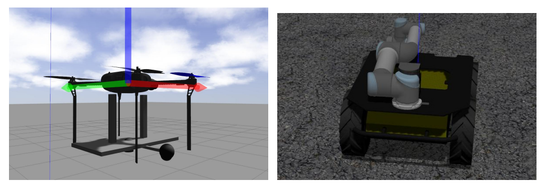

Fig. 6. a) Simulated UAV Model b) Simulated UGV Model

C. Yield Prediction

This module is comprised of two types of machine learning models. The first one is YOLOv5 a deep-learning object detection model to detect and count the visible fruits in the tree. After getting some preliminary fruit count a linear regression model is applied on top of it to accurately map those count with yield. The YOLOv5 model is trained with a mango plantation dataset [ref]. The images are annotated manually with riped fruits and trained for 250 epochs with a batch size of 5 images. The final mean Average Precision (mAP) of the trained model came out to be around 0.943. The model is then used to determine preliminary count of fruits and results of the same can be found below. The next task is to train the regression model with the true yield of each tree and farm.1

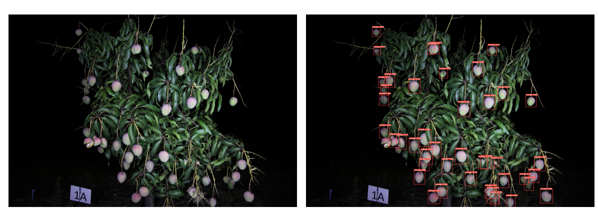

Fig. 7. Yield Prediction: Input Image (Left) Counted Fruits (Right): 41

Fig. 8. Yield Prediction: Input Image (Left) Counted Fruits (Right): 56

D. Drought Stress Detection

In the same simulation environment, the ground is created with some light patches depicting the water-stressed areas. The stress detection pipeline is very similar to the yield predictor. After the action zones are detected, an agent is sent to the location to detect the contour and irrigate it precisely. Stress detection for the farm is tested, and red blobs are drawn to depict the zones (see Fig. 11).

Fig. 9. a) Input Image (Left) b) Generated 3d Point Cloud (Right)

E. Collaborative Mapping

The mapping pipeline has been tested on the images (see Fig. 12a) recorded from a farm near Bangalore, India. The vision module of a master then processes the imagery to generate the point cloud. The point cloud thus generated can be seen in Fig. 12b.

Fig. 10. a) Input Imagery (Left) b) Generated Point Cloud (Right)

V. CONCLUSIONS

The proposed framework can generate, schedule, and monitor the tasks allotted to the agents. The above two cases show how our framework can be deployed in but not limited to agricultural land. It can be deployed in similar scenarios like building construction, mapping, etc. Only the onboard task modules need to be defined for the particular task, as shown in the two cases considered. Simulation results finally confirm the potential of using such a collaborative framework in agriculture.

ACKNOWLEDGMENT

This work is done under the PMRF scheme. We want to acknowledge Mr. P. Chowdappa, who retired as director of Central Plantation Crops Research Institute, Kasaragod, for letting us record and use the data of his farm for our purpose.

REFERENCES

[1] Sylvester, G., “E-Agriculture in Action: Drones for Agriculture,” Tech. rep., Food and Agriculture Organization of the United Nations and International Telecommunication Union, 2018.

[2] Jensen, K., Larsen, M., Nielsen, S. H., Larsen, L. B., Olsen, K. S., Jørgensen, R. N. (2014). Towards an open software platform for field robots in precision agriculture. Robotics, 3(2), 207-234.

[3] Krintz, C., Wolski, R., Golubovic, N., Lampel, B., Kulkarni, V., Sethuramasamyraja, B., ... Liu, B. (2016, August). SmartFarm: Improving agriculture sustainability using modern information technology. In KDD Workshop on Data Science for Food, Energy, and Water.

[4] Katsigiannis, P., Misopolinos, L., Liakopoulos, V., Alexandridis, T. K., and Zalidis, G., ”An Autonomous Multi-Sensor UAV System for Reduced-Input Precision Agriculture Applications,” Mediterranean Conference on Control and Automation, IEEE, Athens, Greece, 2016, pp. 60–64.

[5] Albani, D., IJsselmuiden, J., Haken, R., and Trianni, V., ”Monitoring and mapping with robot swarms for agricultural applications,” 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp.1–6. ”Skyx PASS: Precision Agriculture Spraying Swarm,” Israel, 2018. URL https://www.skyx.solutions/

[6] Bruno S. Faiçal, Fausto G. Costa, Gustavo Pessin, Jó Ueyama, Heitor Freitas, Alexandre Colombo, Pedro H. Fini, Leandro Villas, Fernando S. Osório, Patrícia A. Vargas, Torsten Braun, ”The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides”, Journal of Systems Architecture, Volume 60, Issue 4, 2014, Pages 393-404, ISSN 1383-7621.

[7] Cunha, João P. A. R. da, Sirqueira Neto, Matheus A., Hurtado, Sandro M. C.. (2019). ”Estimating Vegetation Volume of Coffee Crops using Images from Unmanned Aerial Vehicles”. Engenharia Agrícola, 39(spe), 41-47. Epub September 09, 2019.https://dx.doi.org/10.1590/1809-4430-eng.agric.v39nep41-47/2019

[8] Jinru Xue, and Baofeng Su (2017). Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications pp:7-8.

[9] Giakoumidis, N., Bak, J. U., Gomez, J. V. 2012. ”Pilot-Scale Development of a UAV-UGV Hybrid with Air-Based UGV Path Planning”, 10th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, December 17-19, pp 204-208

[10] Park, J.S., Kim, P., Cho, Y., and Fang, Y. (2018).” Automated Collaboration Framework of UAV and UGV for 3D Visualization of Construction Sites.” Proceedings of the 18th International Conference on Construction Applications of Virtual Reality (CONVR2018), Auckland, New Zealand, Nov 22-23.

[11] A. Mukherjee, S. Misra, A. Sukrutha, N. S. Raghuwanshi, ”Distributed aerial processing for IoT-based edge UAV swarms in smart farming”, Computer Networks (Elsevier), Vol. 167, No. 11, Feb 2020, pp. 107038.

[12] Guo, K., Li, X., Xie, L., 2019. ”Ultra-wideband and Odometry-Based Cooperative Relative Localization with Application to Multi-UAV Formation Control”. IEEE Transactions on Cybernetics.

[13] C .Potena, R. Khanna, J. Nieto, R. Siegwart, D. Nardi, and A. Pretto (2019). ”AgriColMap: Aerial-Ground Collaborative 3D Mapping for Precision Farming”, IEEE Robotics and Automation Letters, Vol. 4, No. 2, Mar 2019, pp.1085-1092.

[14] Tantikorn Pichpibula, Ruengsak Kawtummachaib, ”An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem”, ScienceAsia 38 (2012): 307–318.

[15] M Kacira, PP Ling, TH Short. ”Establishing Crop Water Stress Index (CWSI) threshold values for early, non–contact detection of plant water stress”, Transactions of the ASAE, 2002.