A Smart Precision-Agriculture Platform for Linear Irrigation Systems

Jacopo Aleotti, Michele Amoretti, Alessandro Nicoli, Stefano Caselli
Department of Engineering and Architecture
University of Parma, Italy
Email: jacopo.aleotti@unipr.it, michele.amoretti@unipr.it, alessandro.nicoli@studenti.unipr.it, stefano.caselli@unipr.it

***Abstract*—A smart platform is presented that manages the components of an automatic precision irrigation system. The platform has a distributed architecture that includes a decision support system (Irriframe), a server node, a mobile application for user interaction, and embedded IoT devices that operate linear irrigation machines. The decision support system is queried by the server and it computes an irrigation map, i.e., the amount of water to be supplied in each cell of the field by integrating geographic, meteorological and soil data, as well as the vegetation map obtained from an aerial survey and the technical specifications of the irrigation machine. The mobile application is used by the farmer to register user data on the decision support system, to request an irrigation plan from the server and to control the irrigation process with real-time monitoring. Preliminary experiments were conducted in tomato fields to test the main components of the system.**

***Index Terms*—distributed architecture, IoT devices, mobile application, precision irrigation**

I. Introduction

Precision irrigation is an important strategy for water saving based on measuring the variability in space of soil properties and crop responses. Due to the increasing risk of drought in the Mediterranean area, the adoption of precision irrigation technologies is likely to increase. This paper presents a smart platform for precision irrigation that was developed within the ALADIN project funded by the research and innovation program of the region Emilia-Romagna, Italy. The main technical novelty is that the proposed solution is a complete platform that operates linear irrigation machines thanks to a decision support system (Irriframe [1]), which exploits vegetation maps obtained from an aerial survey. The platform has a distributed architecture, including a server node, embedded IoT devices to operate the irrigation machines and a mobile application for user interaction.

Irriframe is a free decision support system for irrigation scheduling. The conventional use of Irriframe is to provide information about when to irrigate as well as the water quantity to be uniformly supplied across the field. Irriframe is based on a water balance model that usually considers geographic, meteorological and soil data.

One of the contributions of this work is to query the decision support system by sending a vegetation map, obtained from an aerial survey, that is used to improve the water balance model as it provides a high resolution measurement of the actual crop vigor. To this purpose an unmanned aerial vehicle (UAV) was equipped with a vision sensor to perform an aerial survey, and software tools were developed for processing the resulting vegetation map.

Therefore, the output of an Irriframe query to the server of the precision irrigation platform is an irrigation map that provides the amount of water to be supplied in each cell of the field rather than a uniform water quantity for the whole field. The detailed irrigation map is then converted to an irrigation plan, i.e., a sequence of machine commands for the irrigation machine deployed on the field. The irrigation process can be controlled and monitored in real-time by the farmer by using the mobile application that receives the irrigation plan from the server.

In the implemented prototype of the smart irrigation platform, the adopted machine is a linear moving system that consists of a hose-fed irrigation boom. Variable rate irrigation is achieved by changing the hose rewind speed. Preliminary experiments are reported that were performed in tomato fields to test the main components of the platform.

The paper is organized as follows. Section II reviews the state of the art. Section III describes the architecture of the smart irrigation platform. Section IV presents the experimental results with the implemented prototype, while Section V concludes the work.

II. Related Work

Most works on variable rate irrigation for precision agriculture investigated the use of fixed sprinklers, while in this paper a fully integrated platform is presented for a linear irrigation system. Moreover, while sensors networks on the ground are usually exploited to compute a water balance and to provide an irrigation schedule [2]–[7], in this work a decision support system is adopted that includes aerial measurements. In [3] a wireless network of moisture and conductivity sensors was developed and tested in several farms. Chikankar et al. [4] presented an automatic irrigation system using ZigBee for power efficiency. In [7] a prototype of a complete context aware wireless irrigation system was developed and evaluated using a single fixed, remotely controllable rotating sprinkler.

The communication between the decision support system and the farmer is a key topic in precision farming. Early works in precision farming adopted the short message service (SMS) [8]–[10], while in this work a mobile application was developed to provide user-friendly interaction.

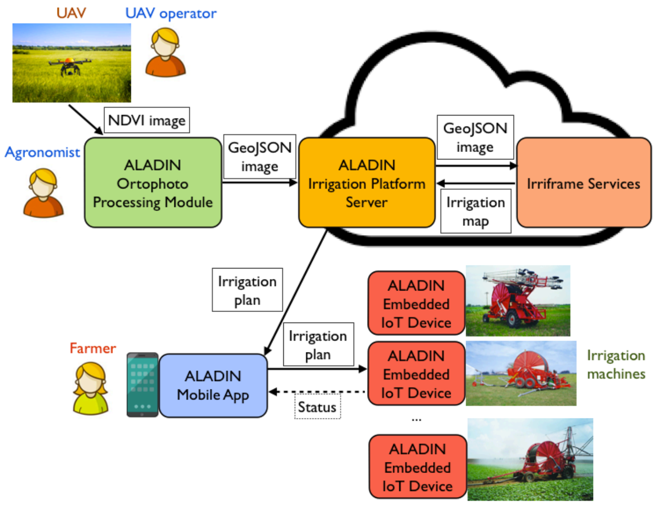

Fig. 1. Architecture of the Smart Irrigation Platform.

To increase the amount of saved water few authors investigated the use of continuous move irrigation systems that support control of individual sprinklers [11]–[13]. While promising, such technologies are still not available in commercial products. In [11] an electronically controllable linear irrigation system was presented with individual control of sprinklers based on a distributed sensor network. In [12] a remote monitoring and control system was developed for a lateral move irrigation system. Camp et. al [13] proposed a modified center pivot system for variable-rate application of water and nutrients.

The use of aerial vehicles for precision agriculture has become widespread [14]–[17]. However, as stated before, integration of UAV data in decision support systems for automated precision farming has not been considered in previous works. A standard technique in UAV surveys is to generate an accurate terrain mosaic from a sequence of multispectral images, and then to compute a vegetation index [15]. Murugan et al. [17] proposed an approach that fuses UAV and satellite images. Thermal sensors can also be adopted in conjunction with multispectral cameras [14].

III. Smart Irrigation Platform

The smart irrigation platform described in this paper is composed by several interacting subsystems, as shown in Figure 1.

A UAV is programmed to fly over the field to be irrigated. The aerial vehicle is equipped with a vision sensor. Geo-localized images taken during the flight are used to produce vegetation maps.

The Ortophoto Processing Module provides a semi-automatic software tool to rotate and to downsample a vegetation map by defining a customized grid. The result is then saved as a GeoJSON file and sent to the Irrigation Platform Server.

The Irrigation Platform Server is the core subsystem, exposing RESTful services to the Ortophoto Processing Module and to the Mobile App. The Irrigation Platform Server also interacts with Irriframe, which accepts GeoJSON files as input and computes an irrigation map that contains the amount of water to be supplied in each cell of the field.

The Mobile App allows end users to request irrigation plans for their machines from the Irrigation Platform Server. Each irrigation plan is a list of commands for an irrigation machine, which is computed by the Irrigation Platform Server from an irrigation map provided by Irriframe.

Further details on the subsystems and functional processes of the smart irrigation platform are provided in the following subsections, which also illustrate the prototype that has been implemented.

A. *Irrigation Platform Server*

The Irrigation Platform Server enables three main processes, namely

1) End User Fields/Machines Mapping,
2) Vegetation Map Upload,
3) Irrigation Plan Computation.

End users authenticate to the platform by means of their Irriframe account. Irriframe may provide the server with the list of fields owned by the end user, but it is not aware of her/his irrigation machines. By means of the Mobile App, the end user inputs a list of irrigation machines, which is stored within the database of the Irrigation Platform Server. At any time, an irrigation machine can be added/removed and associated to a specific field, by means of the mobile app.

The Irrigation Platform Server exposes a RESTful service for uploading a vegetation map exported in GeoJSON file format by the software tool described in Subsection III-C. The software tool itself is provided with client features for invoking the uploading service.

Finally, the farmer is enabled to request the Irrigation Platform Server to compute the irrigation plan, by providing field and irrigation machine details by means of the Mobile App (detailed in Subsection III-B). The Irrigation Platform Server generates the irrigation plan for the specified machine and field, according to the latest irrigation map provided by Irriframe. The irrigation plan is returned to the Mobile App, which forwards it to the embedded IoT device that controls the irrigation machine.

An Irrigation Platform Server prototype was implemented using Node.js [18], a JavaScript runtime based on Google Chrome's engine for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, suitable for data-intensive real-time (DIRT) applications that run across distributed devices. Node.js allows a server to hold a number of connections open while handling many requests and keeping a small memory footprint. It is designed to be responsive, like the browser.

Node.js is also highly modular. Modules can be easily developed, packaged and used. The *npm* repository is an online collection of ready-to-use Node.js modules. For example, the https module was used to provide Node.js with the capabilities of an HTTPS (HTTP protocol over TLS/SSL) server.

The Irrigation Platform Server prototype exposes RESTful services developed by means of the express module for Node.js. With respect to competing modules restify and loopback, express is more efficient [19]. The REST API exposed by the irrigation platform server is available at the following URL: https://aladinirrigation.ddns.net/

Moreover, the Irrigation Platform Server prototype maintains a non-relational database, implemented within the MongoDB [20] document-oriented database management system. In this approach, every document collects all the data associated to an entity, in such a way that any application can treat the entity as a self-contained object and skip the computational burden of aggregating data to extract information related to the same entity, which is typical of relational databases. Such an enormous efficiency advantage is partially reduced by the need to preserve the consistence across partially duplicated documents.

Collections in the aforementioned database have been designed for being as much as possible decoupled, in order to prevent inconsistency issues.

- Users may be either Farmers or Uploaders. Farmers have non-empty lists of fields and machines. Uploaders do not have fields or machines, but are enabled to upload NDVI Maps to the Irrigation Platform Server.
- RasterImgs contains references to vegetation map files which are stored in the file system and accessed by means of the fs Node.js module. Several images may be associated to the same field. They have meaningful name, extension and date (with ms precision).
- Irrigations contains the list of performed irrigations, detailing the farmer, field, tools (irrigation machine and terminals) and date.
- Machines contains irrigation machine descriptions, with parameters such as flow rate, and the list of all acceptable commands (described in Subsection III-D).
- Terminals contains technical information about the terminals that can be associated to the irrigation machines, such as hose-fed irrigation booms.

B. Mobile App

For the farmer, the Mobile App plays the role of single virtual access point to the smart platform. More precisely, the Mobile App allows the farmer to register user data on the decision support system, to request an irrigation plan from the server, and to control the irrigation process with real-time monitoring.

The Mobile App prototype is based on Apache Cordova [21] and Framework 7 [22]. Apache Cordova is a framework for developing mobile apps with HTML, CSS and JavaScript, targeting multiple platforms with one code base. It is free and open source. Framework 7 is an HTML5 framework simplifying the development of hybrid applications with native look and feel, and an effective prototyping tool.

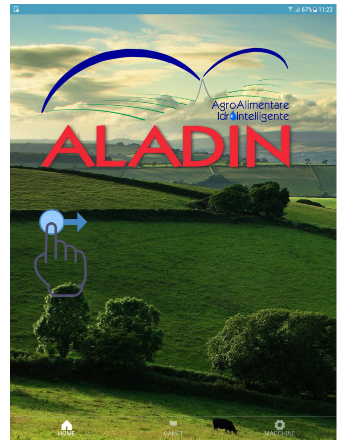

Fig. 2. Mobile App: login view.

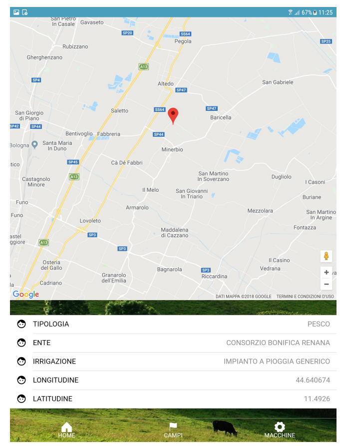

Fig. 3. Mobile App: details of the selected field.

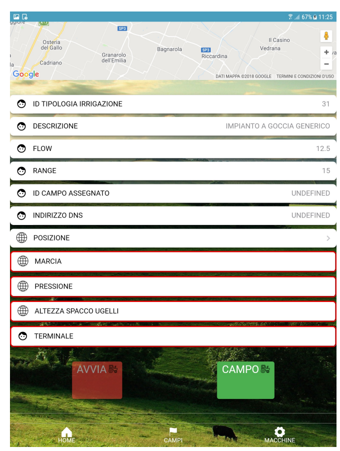

Fig. 4. Mobile App: configuration of the smart irrigation process.

The implemented Mobile App has a Single View layout,

Fig. 5. Unmanned aerial vehicle used for surveying and mapping crop fields.

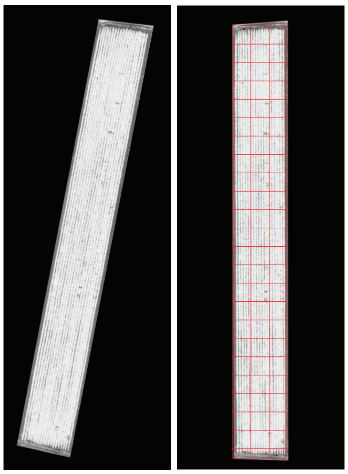

Fig. 6. Raw NDVI vegetation map (left) in a tomato field. Rotated image (right) with superimposed grid (red lines).

with Tab-bar, Nav-bar and side panels for navigation purposes. The main advantage of this approach is that content is fully loaded when the application is started, for enhanced responsiveness and improved user experience.

Since part of the visualized information comes from Irriframe, to start the Mobile App the Farmer is required to perform a login process using Irriframe credentials (Figure 2). After login, it is possible to enter the panels for managing the farmer's fields and irrigation machines.

The Fields panel lists the fields owned by the farmer. For each field, it is possible to visualize the coordinates of the centroid of the field, the farmer's company name (if any), the type of crop and the adopted irrigation approach (Figure 3). On the map, the borders of the selected field are also displayed.

The Machines panel lists the irrigation machines owned by the farmer. For each machine, it is possible to know the current location. Moreover, it is possible to start the smart irrigation process, upon specifying some mandatory input parameters, such as pressure, nozzle cut height, gear (Figure 4). Finally, it is possible to monitor the elapsed time and the percentage of completion of all ongoing processes.

C. Integration of vegetation map from aerial imagery

After each flight of the UAV, all collected images are processed (by the Ortophoto Processing Module) to create orthophotos of the RGB bands of the camera. Using a single

Fig. 7. The irrigation machine that was used to test the smart irrigation platform. The total width of the wings is 40 m.

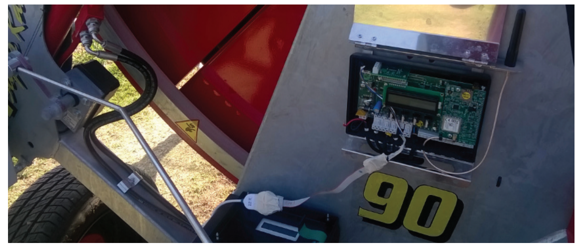

Fig. 8. Embedded IoT device and irrigation machine electronic control unit (ECU).

camera, instead of more advanced multispectral sensors, still enables the computation of a vegetation index although it slightly degrades its sensitivity. Indeed, a red band cut-off filter is used in place of the standard NIR filter of the camera. Therefore, the red channel contains the NIR (Near-infrared) radiation. A vegetation map (in GeoTIFF format) of the field is then computed that contains for each pixel the NDVI (Normalized Difference Vegetation Index) index, defined as

$$NDVI = \frac{NIR - BLUE}{NIR + BLUE} \tag{1}$$

Figure 6 (left) shows an example of an NDVI vegetation map obtained in a tomato field in gray scale.

A semi-automatic software tool was developed to rotate the raw NDVI vegetation map so that the crop rows are aligned with the vertical y-axis of the image. The algorithm extracts the principal direction of the crop rows by computing the Hough transform of the image. If the algorithms fails to detect the correct rotation angle a manual correction can be inserted by the user. Figure 6 (right) shows the NDVI vegetation map of the example tomato field obtained after automatic rotation. The software tool also allows to perform average downsampling of the NDVI vegetation map by defining a grid of cells (red grid in Figure 6, right) at a lower-resolution. The resolution of the grid can be selected according to the actual performance of the irrigation machine (about 10 m resolution in the current setup). The result is exported to GeoJSON file format.

D. Automatic programming of irrigation machine

The linear irrigation machine considered in this work is a hose-fed irrigation boom. The one that was actually used to test the smart irrigation platform prototypes is shown in Figure 7. Since the water pressure at each sprinkler is constant, variable rate irrigation is achieved by changing the hose rewind speed of each horizontal section of the field as explained next. The irrigation machine is automatically programmed by an embedded IoT device designed to execute and to monitor the irrigation plan.

The irrigation machine used for testing purposes is controlled at low level by a Rain 260 electronic control unit (ECU), by RM Irrigation Equipment S.p.A. The ECU accepts commands to start and to stop the irrigation, to set the rewind speed, to read the length of the unrolled hose, and to pause irrigation in case of damage. The embedded IoT device consists of a Raspberry Pi3 with a GPS receiver, and runs a software based on Redis [23], an open-source in-memory key-value framework. The embedded IoT device is connected to the ECU through a RS-232 cable, and it also communicates with the Mobile App and the Irrigation Platform Server through a 4G network connection. The embedded IoT device and the irrigation machine electronic control unit are shown in Figure 8.

All commands contained in an irrigation plan are stored in a Redis queue and executed sequentially. Each command specifies how to irrigate a section of the field. In particular, a command consists of a pair (v_i, l_i), where v_i is the desired rewind speed set-point, and l_i is the length of the i-th section of the field. The speed value v_i is set as inversely proportional to the average water quantity to be supplied to the i-th section, i.e.

$$v_i = \frac{k}{\frac{1}{N_i}\sum_{j=1}^{N_i} w_{ij}} \tag{2}$$

where w_{ij} is the amount of water in the irrigation map to be supplied to cell (i, j), N_i is the total number of cells in section i and k is a constant value that depends on internal machine parameters (water pressure, boom length, nozzle cut height and gear). A control loop in the embedded IoT device monitors in real-time the position of the boom in the field by polling the current length of the unrolled hose from the ECU. If the field width is greater than the length of the boom, irrigation must be stopped at the end of the current lane and restarted, after repositioning the boom at the beginning of the next lane. The GPS signal is used before starting the irrigation of each lane of the field to check whether the initial position of the boom is correct.

IV. Experimental evaluation

We report preliminary experiments that were carried out to test the main components of the system separately. A first experiment was conducted for the aerial system.

The adopted aerial vehicle is shown in Figure 5. The UAV is a VirtualRobotix SPARK quadcopter (340 × 340 mm size, 190 mm height) with a payload of about 300 g. The estimated autonomy is up to 22 minutes. The UAV is powered by a VR Brain 5.2 board and it features a composite endoskeleton with power distribution features. The UAV is equipped with a Raspberry Pi3 computer and a Raspberry Pi NoIR camera

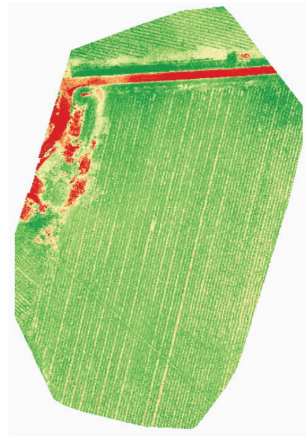

Fig. 9. NDVI image generated after a test flight in a tomato field.

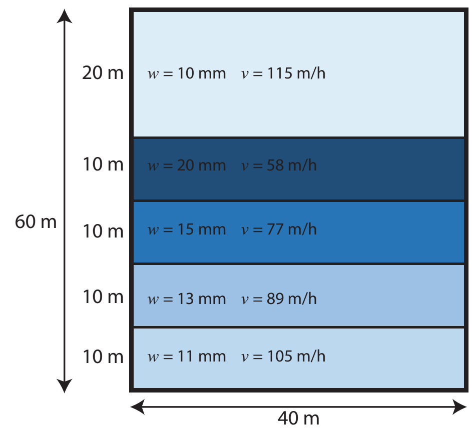

Fig. 10. Irrigation plan for a test field 60 × 40 m (Marzolara, Bologna). The scheduled amount of water w and the rewind speed v are reported for each section of the field. Irrigation direction is from bottom to top.

Fig. 11. Image from the irrigation test described in Figure 10.

(V2). The camera has a Sony IMX219 CMOS sensor (8 megapixel) and it features a maximal resolution of 3280×2464 pixels (30 fps). Radiometric calibration is performed by using a reference target.

The UAV was programmed to perform a flight mission over a tomato field (100 × 150 m) in Vigheffio (Parma). A total of 73 images were acquired and the obtained NDVI orthophoto is shown in Figure 9. The NDVI values range from 0.25 (red) in non-crop regions, like in a nearby road, to 0.39 (green) for vegetation. The experiment indicates that the aerial system is capable of generating plausible NDVI images, even though

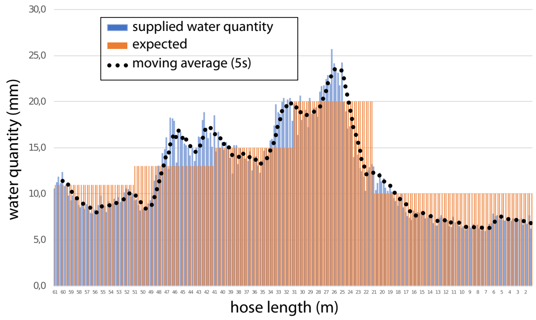

Fig. 12. Actual vs expected amount of water supplied to the field in the test described in Figure 10, from 60 m (start) to 0 m (end).

with limited sensitivity. Moreover, being the test field well vegetated the resulting NDVI image exhibits a low-variance on the crop region. The result was not unexpected as the specific test field available for the project experimentation had a limited spatial heterogeneity of soil.

Evaluation of the automated irrigation process was performed in a tomato field (60 m $\times$40 m) in Marzolara (Bologna) using a custom-built irrigation map. The irrigation map was automatically converted by the platform server to an irrigation plan that consists of five sections of different length, as shown in Figure 10. Then, the irrigation was started by using the mobile application. The top speed was about 115 m/h. Figure 11 shows a picture taken during the irrigation test. The graph in Figure 12 illustrates a comparison between the actual amount of water that was supplied to the field, measured by rain gauges, and the expected one. Although an induced delay can be observed due to the hydraulic system, it can be noticed that the supplied amount of water follows the expected value.

V. Conclusion

In this work a distributed smart precision agriculture platform was presented for a linear irrigation system. The platform exploits a decision support system that generates an irrigation map by integrating geographic, meteorological and soil data as well as a vegetation map computed from an aerial survey. The irrigation map is converted to an irrigation plan, i.e., a list of machine commands. The irrigation process can be started and monitored by the farmer thanks to a mobile application. Preliminary experiments have been reported in tomato fields. Additional experiments are planned to evaluate more thoroughly the proposed platform. Future work will address the online optimization of the irrigation process based on water supply monitoring performed on the irrigation machine.

Acknowledgment

This research was supported by the ALADIN collaborative project (European Regional Development Fund POR-FESR 2014-2020, Research and Innovation of the Region Emilia-Romagna, CUP D92I16000030009). The authors would like to thank AeroDron S.r.l., RM Irrigation Equipment S.p.A and Startec Italy S.r.l. The authors would also like to thank all the partners of the ALADIN project.

References

[1] P. Mannini, R. Genovesi, and T. Letterio, “Irrinet: Large scale DSS application for on-farm irrigation scheduling,” *Procedia Environmental Sciences*, vol. 19, pp. 823 – 829, 2013.

[2] S. Fazackerley, A. Campbell, R. R. Trenholm, and R. Lawrence, “A holistic framework for water sustainability and eduction in municipal green spaces,” in *25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, April 2012, pp. 1–6.

[3] R. Zaier, S. Zekri, H. Jayasuriya, A. Teirab, N. Hamza, and H. Al-Busaidi, “Design and implementation of smart irrigation system for groundwater use at farm scale,” in *7th International Conference on Modelling, Identification and Control (ICMIC)*, Dec 2015, pp. 1–6.

[4] P. B. Chikankar, D. Mehetre, and S. Das, “An automatic irrigation system using zigbee in wireless sensor network,” in *International Conference on Pervasive Computing (ICPC)*, Jan 2015, pp. 1–5.

[5] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, “Energy efficient automated control of irrigation in agriculture by using wireless sensor networks,” *Computers and Electronics in Agriculture*, vol. 113, pp. 154 – 163, 2015.

[6] A. J. Phillips, N. K. Newlands, S. H. Liang, and B. H. Ellert, “Integrated sensing of soil moisture at the field-scale: Measuring, modeling and sharing for improved agricultural decision support,” *Computers and Electronics in Agriculture*, vol. 107, pp. 73 – 88, 2014.

[7] D. Sonkiya, P. Sonkiya, P. Das, M. V. V., and M. V. Ramesh, “Cawis: Context aware wireless irrigation system,” in *International Conference on Computer, Communications, and Control Technology (I4CT)*, Sept 2014, pp. 310–315.

[8] N. Car J., E. Christen, J. Hornbuckle, C. Bartlett, and G. Moore, “Tools for improving water use efficiency: Irrigation informatics implemented via SMS,” in *22nd International Congress on Modelling and Simulation (MODSIM2017)*, 2007, pp. 671–677.

[9] A. Singels and M. T. Smith, “Provision of irrigation scheduling advice to small-scale sugarcane farmers using a web-based crop model and cellular technology: a south african case study,” *Irrigation and Drainage*, vol. 55, no. 4, pp. 363–372, 2006.

[10] N. Inman-Bamber, S. Attard, S. A. Verrall, W. A. Webb, and C. Baillie, “A web-based system for scheduling irrigation in sugarcane,” in *Proc. Int. Soc. Sugar Cane Technol.*, vol. 26, 01 2007.

[11] Y. Kim, R. G. Evans, and W. M. Iversen, “Remote sensing and control of an irrigation system using a distributed wireless sensor network,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1379–1387, July 2008.

[12] J. L. Chávez, F. J. Pierce, T. V. Elliott, and R. G. Evans, “A remote irrigation monitoring and control system for continuous move systems. part A: description and development,” *Precision Agriculture*, vol. 11, no. 1, pp. 1–10, Feb 2010.

[13] C. Camp, E. Sadler, D. Evans, L. Usrey, and M. Omary, “Modified center pivot system for precision management of water and nutrients,” *Applied Engineering in Agriculture*, vol. 14, 1998.

[14] J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres, “Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, March 2009.

[15] J. Navia, I. Mondragon, D. Patino, and J. Colorado, “Multispectral mapping in agriculture: Terrain mosaic using an autonomous quadcopter UAV,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 1351–1358.

[16] M. Ponti, A. A. Chaves, F. R. Jorge, G. B. P. Costa, A. Colturato, and K. R. L. J. C. Branco, “Precision agriculture: Using low-cost systems to acquire low-altitude images,” *IEEE Computer Graphics and Applications*, vol. 36, no. 4, pp. 14–20, July 2016.

[17] D. Murugan, A. Garg, and D. Singh, “Development of an adaptive approach for precision agriculture monitoring with drone and satellite data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, pp. 5322–5328, Dec 2017.

[18] “Node.js Home Page,” https://nodejs.org.

[19] “Node.js performance 2017: v7.9.0 vs. Hapi, Express.js, Restify and Koa and more,” https://raygun.com/blog/node-js-performance-2017.

[20] “MongoDB home page,” https://www.mongodb.com.

[21] “Apache Cordova Home Page,” https://cordova.apache.org.

[22] “Framework 7 Home Page,” https://framework7.io.

[23] “Redis Home Page,” https://redis.io/.