Computers and Electronics in Agriculture 154 (2018) 289–295

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

An open-source spatial analysis system for embedded systems

Andre Luiz de Freitas Coelho*, Daniel Marçal de Queiroz, Domingos Sárvio Magalhães Valente, Francisco de Assis de Carvalho Pinto

Departamento de Engenharia Agricola, Universidade Federal de Vicosa, Av. PH Rolfs, s/n, 36570-900 Vicosa, Minas Gerais, Brazil

ARTICLE INFO

Keywords:
Precision agriculture
Ordinary kriging
Clustering analysis
Yield map
Data filtering

ABSTRACT

Soil and plant monitoring systems are important tools for applying precision agriculture techniques. To acquire soil-plant system data, the user establishes a sampling strategy, goes to the field, collects data and finally goes to the office for data analysis. Sometimes, when the analysis is performed, the user realizes that the sampling strategy was not adequate and needs to return to the field in order to collect more data. To avoid problems with the sampling strategy, the solution is to have a system that performs the data analysis immediately after its collection, while the user is still in the field. To do that, we can use single board computers; these types of platforms have ports to communicate to sensors and good processing capabilities. Therefore, the objective of this work was to develop an embedded system to perform spatial variability data analysis in the field, right after data acquisition. The software was developed using Python 3.6; the PyQt Integrated Development Environment (Riverbank Computer Limited, Dorchester, United Kingdom) was used to design a graphical user interface. The BeagleBone Black board, running Debian version 8.6, was used to implement the software. The analysis was divided into three steps: in the first one, an outlier and inlier analysis was performed to remove unwanted data; in the second one, the semivariogram was generated, and the variable and standard deviation map was produced by performing ordinary kriging; and in the last one, a cluster analysis was performed to create management classes using a fuzzy k-means algorithm. The graphical user interface showed the variable map and the variable classes map. To test the developed software, soybean yield data that was collected in a 31.6-ha field were used. The developed software was shown to be efficient at performing the spatial variability of soybean yields. The comparison of the generated maps shows the importance of filtering the data before performing the analysis. The developed software is available at the GitHub website.

1. Introduction

Plant and soil monitoring are routine tasks when adopting precision agriculture techniques (Queiroz et al., 2017). The monitoring systems are composed of at least a sensor system and a microprocessor for data acquisition and/or visualization. In these systems, the user generally defines the sampling strategy and goes to the field with the system to acquire and/or store the desired data. The field scan can be performed manually or with the help of land or air vehicles.

After the data are collected, they are transferred to a computer in an office to be analyzed so they can be used to define the following field management operations. However, errors can be detected during the analysis that affect the accuracy of the spatial variability representation of the variables that were measured. These errors may be related to inadequate sampling, failure of sensors due to calibration or human mistakes during system operation. If a problem is detected, it is recommended to return to the field and collect more data. This task generates more cost and is time-consuming. Therefore, it is beneficial to avoid this kind of problem. Another problem is that, depending on the type of data collected, a large amount of incoherent data are acquired, which makes it difficult to analyze in the office. A possible solution is to develop a system with embedded software capable of collecting sensor data, removing incoherent data and performing the analysis. In this case, only useful data are stored.

Single board computers, such as BeagleBone Black (BBB), are a good alternative to be employed in the development of monitoring systems for precision agriculture. BBB has ports to connect sensors and has processors capable of performing complex calculations (Molloy, 2014). Due to these characteristics, it has been used in the development of several systems to collect data related to agriculture. Queiroz et al. (2017) developed a portable sensor for the determination of soil apparent electrical conductivity using a BBB. The developed system had software, written in C++, capable of performing geostatistics analysis using ordinary kriging interpolation. The analysis of the standard

* Corresponding author.
E-mail addresses: andre.coelho@ufv.br (A.L.d.F. Coelho), queiroz@ufv.br (D.M. de Queiroz), valente@ufv.br (D.S.M. Valente), facpinto@ufv.br (F.d.A.d.C. Pinto).

https://doi.org/10.1016/j.compag.2018.09.019
Received 19 February 2018; Received in revised form 11 September 2018; Accepted 14 September 2018
0168-1699/

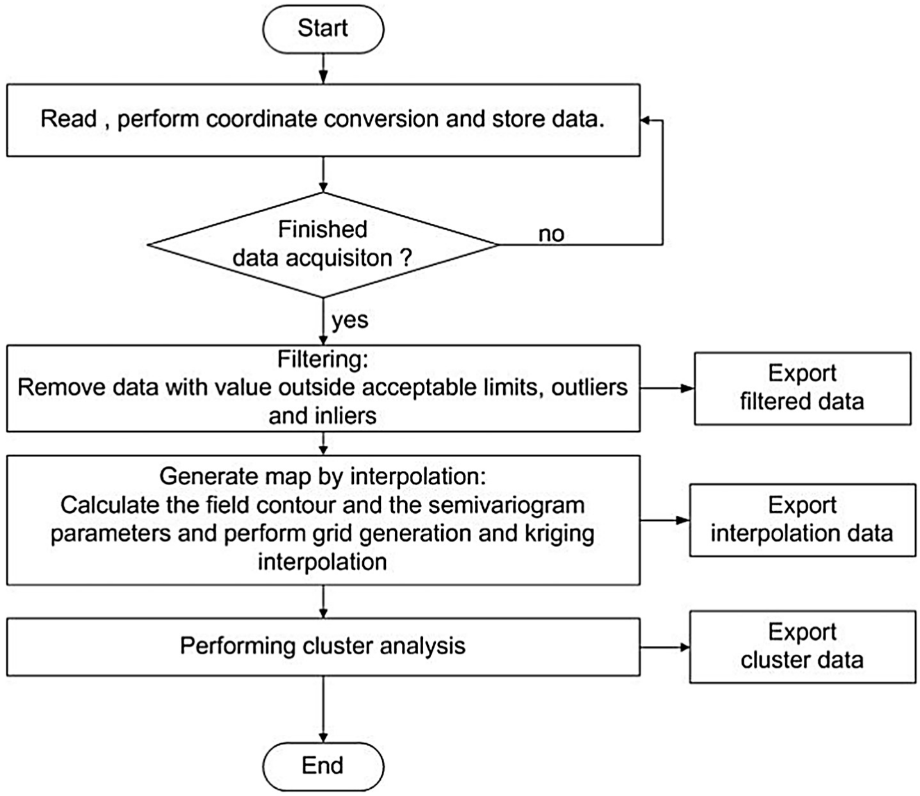

Fig. 1. Flowchart of the developed software.

deviation was used to predict the position of new points to be sampled in order to obtain greater precision in the characterization of the spatial variability of the soil apparent electrical conductivity. Fisher and Huang (2017) developed a plant canopy monitoring system using only low-cost components. The system was developed to store data in a SD card, allowing the user to perform data analysis in the office with software able to perform geostatistics analysis. With the objective of collecting data without performing any geostatistics analysis, Bitella et al. (2014) developed a low-cost sensor for monitoring soil water content and multiple soil-air-vegetation parameters. Following the same path, Thomasson et al. (2006) used a single board computer to develop a yield monitor for a peanut harvesters based on an optical sensor.

On the other hand, open source computer programs and protocols are available for data filtering (Simbahan et al., 2004; Spekken et al., 2013; Sudduth and Drummond, 2007; Taylor et al., 2007), for ordinary kriging interpolation and for clustering analysis, such as Vesper (Whelan et al., 2002), FuzMe (Minasny and McBratey, 2002) and KrigMe (Valente et al., 2012). Thus, this work proposes an open source embedded system capable of acquiring data and performing data filtering, spatial variability analysis and cluster analysis of the collected data.

2. Materials and methods

The software was developed using Python language version 3.6; its graphical user interface was generated using PyQt5 (Riverbank Computer Limited, Dorchester, United Kingdom). The single board computer that was used to run the software was the BeagleBone Black (BBB), Rev B, with the Linux operating system Debian 8.6. The BBB is based on a 1 GHz Sitara AM3358 ARM Cortex-A8 1 GHz processor with 512 MB ram memory. It integrates rich peripheral interfaces, including HDMI, USB, SD card slots and 96 GPIO (general purpose input/output) ports. The user interface is acquired by an 800x480 pixel touchscreen LCD cape that is connected to the BBB and displays to the user every nuance of the developed software.

The software was developed to perform data acquisition and geostatistical analysis in three steps (Fig. 1). The data acquisition process consisted of reading the data from the sensors, performing the conversion of the latitude-longitude coordinate system into Universal Transversal Mercator (UTM) coordinates, and storing the data and coordinates in the memory card of the BeagleBone Black board. For the conversion to UTM, the WGS84 (World Geodetic System 84) datum was used. The geostatistics analysis was composed of data filtering, generation of the variable and standard deviation maps by ordinary kriging and generation of the class map by clustering analysis.

After finishing data acquisition, the first function assigned to the software was to filter the data collected by the sensors. The program was developed based on the procedure proposed by Taylor et al. (2007), where the filtering is divided into three steps: removal of points exceeding minimum and maximum acceptable values, removal of outliers and removal of inliers. In the first step, the user needed to establish a bottom and a top threshold based on his/her knowledge about the behavior of the data collected.

In the outlier removal step, the program calculated the mean and the standard deviation (SD) of the dataset, excluding data outside boundaries that were set in the previous step. The point candidates for removal were those higher than m times SD above the mean or m times SD below the mean. Here, m is a standard deviation factor defined by the user, the default value of which was stablished as 2.5 (Taylor et al., 2007). The computer program was developed in such a way that before the candidate point was completely removed from the dataset, its value was compared with the median of the neighboring values that were no more than 25 m away from the point being analyzed. Points that did not