A method to improve the performance of raster selection based on a user-defined condition: An example of application for agri-environmental data

Driss En-Nejjary, François Pinet, Myoung-Ah Kang

Author-produced version of the article published in Advances in Intelligent Systems and Computing, 2019, 893, 190-201. The original publication is available at https://www.springer.com/ doi : 10.1007/978-3-030-04447-3_13

A Method to Improve the Performance of Raster Selection Based on a User-defined Condition: an Example of Application for Agri-environmental Data

Driss EN-NEJJARY[1,2], François PINET[2] and Myoung-Ah KANG[1]

[1] LIMOS
University Clermont-Auvergne,
Campus Universitaire des Cézeaux,
1 rue de la Chebarde, TSA 60125, CS 60026,
63178 AUBIERE CEDEX – France

driss.en-nejjary@irstea.fr
kang@isima.fr

[2] Irstea, TSCF,
Research Unit "Technologies and Information Systems for Agricultural System",
Clermont-Ferrand Centre,
9 avenue Blaise Pascal, CS 20085,
63178 Aubière

francois.pinet@irstea.fr

Abstract. More and more environmental and agricultural data are now acquired with a high precision and temporal frequency. These data are often represented in the form of rasters and are useful for agricultural activities or climate change analyses. In this paper, we propose a new method to process very large raster. We present a new technique to improve the execution time of the selection and calculation of data summaries (e.g., the average temperature for a region) on a temporal sequence of rasters. We illustrate the use of our approach on the case of temperature data, which is important information both for agriculture and for climate change analyses. We have generated several data sets in order to analyze the influence of the different value properties on the process performance. One of our final goals is to provide information about the value conditions in which the proposed processing should be used.

Keywords: Agri-environmental data, Raster selection, Data processing

1 Introduction

The volume of environmental data becomes very important. More and more environmental and agricultural data are now acquired automatically at high precision and temporal frequencies [1-6]. Numerous data sources are available in different information systems [7] and can be accessed through the Web [8-11]. Several of these data sets are useful for agricultural activities or climate change analyses. They can be related to weather, sensor measurements, soil condition, etc. These types of information can be used in agriculture for example for recommendation on the use of agricultural inputs (water, phytosanitary treatments, etc.) or for crop management in order to optimize and reduce the use of agro-equipments, and consequently the negative impact on climate. These data can also be utilized to analyze the links between different agricultural activities (livestock, crops, etc.) and the climate change, at a large spatial and temporal scale.

Several of these large data sources are represented in the form of rasters, e.g., geo-referenced regular grids [12-13]. This type of 2-dimmensional grids constitutes a traditional geographical data format. In geographical information systems, a raster is a 2-D matrix of cells. A measurement (which is very often a numeric value) is stored in each raster cell to represent the geo-referenced value of environmental phenomena: temperature, soil moisture, CO2 measurements, rain precipitation, etc. This type of data can also be produced by simulation.

Dedicated methods are needed to manage the huge volume of rasters produced over time. It is important to propose specific techniques to optimize the analysis and the processing of such data sources.

In this paper, we propose a new method to analyze large sets of rasters. Our goal is to propose a new technique to improve the execution time of the selection and calculation of data summaries (e.g., the average temperature for a region) on a temporal sequence of rasters. We illustrate the use of our approach on the case of temperature data, which is crucial information both for agriculture and for climate change analyses.

Section 2 describes more precisely the raster data process discussed in this paper and highlights its interest. Section 3 introduces our heuristic to improve the performance in terms of processing time. Sections 4 and 5 show some first experiments of this method on simulated data. Section 6 presents conclusion and future work.

2 Raster data process

The raster data process used in this paper consists in three main steps (shown in Figure 1). The process is based on a large sequence of rasters, representing the evolution of an environmental pheromone over time. In Figure 1, the different values of the raster cells are represented by colors. In the step (a), the user chooses a period of interest. More precisely, he/she selects a temporal raster (sub)sequence of interest in the large sequence of rasters. In the step (b), the user defines the geographical region to analyze in the sequence of rasters selected in step (a). This geographical region to analyze is the same for all these rasters. In the step (c), the system automatically selects every raster that satisfies a user-defined condition. Figure 2 provides a UML representation of the different object types.

We illustrate this process on an example. A user would like to analyze a sequence of rasters representing the evolution of temperatures. He/she wants to determine the set of rasters having low temperatures in order to:

- study more precisely these cases and their possible local causes. It is a typical case of climate change analysis.
- or analyze the impact of these temperature on crops in agriculture in the context of farm decision support.

First, he/she manually chooses the period to be analyzed in the whole sequence (step (a)). Second, he/she manually chooses a geographical region of interest for his/her study (step (b)). Third, in the step (c), the user would like to automatically select every raster in which the average temperature of the region of interest is lower than a user-defined threshold (e.g., ≤ 10 C°). Consequently, the result is the set of the rasters that satisfy this condition.

In our scenario, the steps (a) and (b) imply a manual user choice. The step (c) can be automated, e.g., the calculation of the average temperature for every raster and the selection of the temperature $\leq$ 10C°. A naive algorithm for the step (c) is shown in Figure 3:

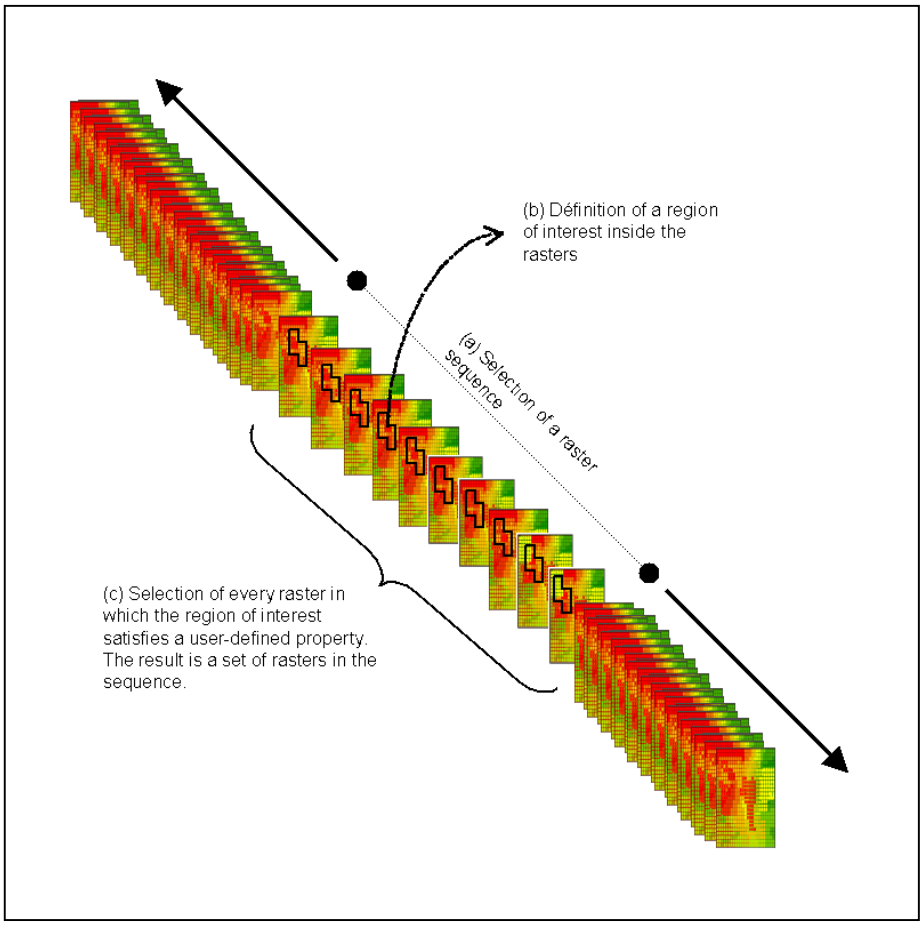

Fig. 1. Description of the raster data process.

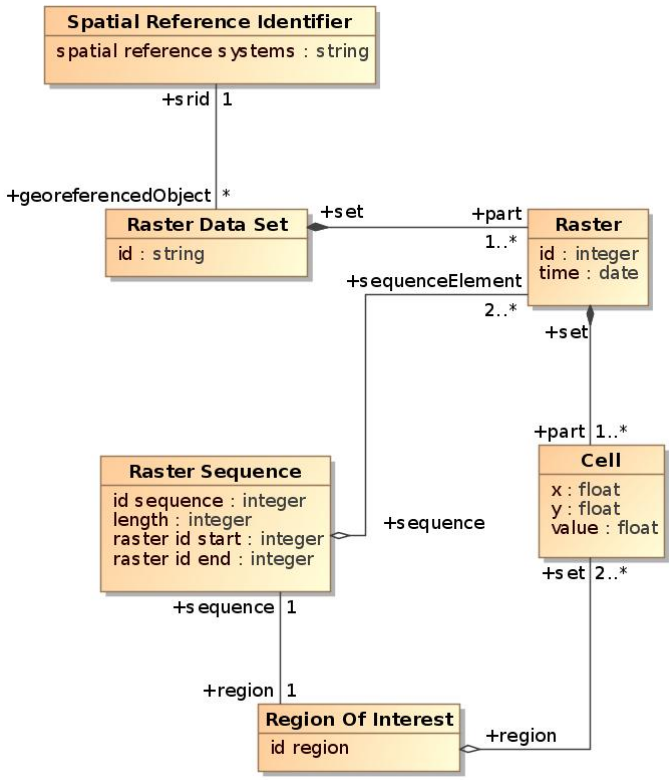

Fig. 2**.** Raster data set modelling.

```
S := the whole sequence of rasters
A := the (sub)sequence of rasters chosen in S by the user
b := the region of interest chosen by the user/
Result := { }

for every raster Ri in A
{
    avg := the cell average for the region b in Ri
    if avg < 10 then Result := Result ∪ { Ri }
}
```

Fig. 3**.** Naive algorithm for step (c)

3 Raster data process

In the present paper**,** we propose a method to improve the performance for the step (c). The intuition behind this algorithm is to try to reject a raster that does not satisfy the user-condition (i.e., the cell value average must be lesser then the user's threshold) as soon as possible to avoid useless computation. The proposed technique can improve the computation when the user's threshold is low (compared to the raster cell values). In this technique, the cell must contain only numerical positive value – consequently, a uniform translation or normalization must be used if the rasters do not comply with this constraint.

The calculation of the average is computed for each raster (in the region of interest). The average computation consists in calculating the sum of cell values for each raster. In the new version of our algorithm, we stop the sum computation as soon as possible, when we are sure that this sum becomes superior to the threshold value multiplied by the cell number of the region of interest.

We also propose to sort the cell values in the region of interest in a descending order, for the average computation. In that case, the threshold is reached faster for the rasters that do not satisfy the condition. Unfortunately, the time complexity of a sort, i.e., $O(n \log n)$ for a quick sort, is higher than the sum computation, i.e., $O(n)$. Consequently, we propose the following stages:

1) We propose to sort the value of the region of interest only for some rasters, e.g., compute a sort every 200 rasters, in sorting the cell values of the region of interest only for the rasters R_i, R_{i+200}, R_{i+400}, etc. Each one of these sorts produces a cell ordering.

2) We propose to use the cell ordering of the sorted rasters, for computing the sums for the other rasters. For example, the sort in R_i produces a cell ordering. This cell ordering will be used for computing the sum for each raster from R_i to R_{i+199}. The cell ordering determined by the sort of R_{i+200} will be used for each raster from R_{i+200} to R_{i+399}, etc.

The intuition behind this method is that in many phenomena the spatial distribution of values evaluates rather slowly over time. In the case of temperature rasters produced every 5 minutes, the highest values will often be on the same geographical part of the rasters for several tens of minutes or several hours. The frequency of the sort computation can be adapted to the nature of the data (e.g., sorting every 10 rasters, 50 rasters, 100 rasters, 200 rasters, etc.). This new version of the algorithm is shown in Figure 4. In Figure 4, *Ord* is an array that corresponds to a mapping: *Ord(1)* is equal to the cell number (#) in *b* that has the highest value; *Ord(m)*is equal to the cell number (#) in *b* that has the lowest value.

Several constraints must be satisfied in order to guarantee that this algorithm provides better performances in terms of execution time, for example, a low user-defined threshold or a spatial distribution of cell values sufficiently large in every raster to justify the interest of the sorting operation. These aspects and several proposed improvements are discussed in the last section of the paper.

4 Experiment description

To evaluate the performance of our improved algorithm of raster selections, we ran our experiments on the Intel(R) Core(TM) i5-5350U CPU at 1.8 GHz. Concerning the data, the experiments are conducted using a public data set.

We used the public dataset provided by the US National Oceanic and Atmospheric Administration [14]. This dataset provides a large amount of climate and historical weather data, such as: air temperature, humidity, precipitation, etc. available in different temporal acquisition frequencies: monthly, daily, hourly and sub-hourly (5-minute). These data are produced from many weather stations in the USA and elsewhere. In our work we have used the daily frequency of data acquisition for four years 2014, 2015, 2016 and 2017 of the Harrison station in USA. Each day, we have the min, max and the average of the temperature of this latter. We have chosen the daily temperature to have a significant difference between the minimum temperature and the maximum temperature for the station, the thing that we need in our algorithm, which is not the case for the hourly data for instance. In order to obtain a large and significant data set, we have generated (i.e., simulated) rasters from the station data.

Improved algorithm for step (c):

S := the whole sequence of rasters
A := the (sub)sequence of rasters chosen in *S* by the user
b := the region of interest chosen by the user
m := the number of cells in *b*
begin := the number (#) of the first raster in *S*
end := the number (#) of the last raster in *S*
it := the interleave between two consecutive rasters on which a sort is calculated
th := the user-defined threshold

Result := { }
maxsum := *th* * *m*

for *i* := *begin* to *end* step *it*
{
    Sort the cell values of the region *b* in R_i in descending order
    and produce the corresponding cell ordering *Ord*

    for *j* := *i to i+it*
    {
        if *j* > *end* then { process completed ; stop } else
        {
            su := 0
            reject := *false*
            for *k*:= 1 to *m*
            {
                v := the value in the $cell_{Ord(k)}$ in R_j
                su := *su* + *v*
                if *su* > *maxsum then* { *reject* := *true* ; *break*}
            }
            if *reject is* false then *Result* := *Result* ∪ { *Rj* }
        }
    }
}

Fig. 4. Improved algorithm for step (c)

To build our dataset, we have simulated rasters for the local studied region. We assumed that the temperature of a region has a Gaussian distribution. The normal distribution has two characteristics: the mean and the standard deviation. The mean is included in the initial dataset, and we have estimated the standard deviation *ST* using the min and the max values (also provided by the initial dataset). The simple method to simulate the standard deviations *ST* is the range rule of thumb [15]. Here is the calculation of the approximation of *ST*:

$$ST \approx \frac{max - min}{4}$$

In practice, the estimation of *ST* using the range rule of thumb is not sufficient when the *n* is extremely small or large. The authors of [15] have improved this estimation in order to deal with this problem:

$$ST \approx \begin{cases} \frac{1}{\sqrt{12}}\left[(max - min)^2 + \frac{(max - 2m + \min)^2}{4}\right]^{1/2} & n < 15 \\ \frac{max - min}{4} & 15 < n < 70 \\ \frac{max - min}{6} & n > 70 \end{cases}$$

In our experiment, we have simulated a large data set. Consequently, we used the third case for the estimation of our standard deviation (n >70). Thus, we had the required parameters to produce raster data from the initial data using a normal distribution.

We have created 3 data sets having 3 different raster sizes; each data set contains 1420 rasters. We have applied a translation on the cell values in order to avoid negative value; we have added a constant (50) to each cell value. In our tests, the user-defined region of interest is the whole raster. In the produced data sets, we have one raster every day for four years. The tests have been applied on all these rasters – these rasters constitute the sequence *A* of rasters to analyze.

5 Results

In this section, we show the results of our experiments performed on the generated data. Different raster sizes have been tested. For each experiment, we test the naive algorithm and the improved version on the same data set. In our experiments, we also

evaluate the impact of the main parameters on the execution time of our algorithm, for instance, the threshold and the interleave between the sorted rasters. To do this, we have chosen different thresholds and interleaves and run our algorithm using these different value parameters. Concerning the sort algorithm, we used a quick sort.

5.1 The impact of the threshold on the performance

Tables 1, 2 and 3 compare the computing time for the naive and the improved algorithm for the three data sets for different threshold values.

Table 1 shows that the improved algorithm is faster than the naive one, especially when the threshold is not too low and not too high. The best performance is with *th*=40; our algorithm I is faster than the naive one with 3.07 seconds less for time execution. Whereas when the threshold is smaller, we obtain less performance (the case of *th*=30).

Table 1: Dataset 1: Size of raster =100×100, contains 1420, Interleave =73

	Threshold *th*=30	**Threshold *th*=40**	**Threshold *th*=45**	**Threshold *th*=46**	**Threshold *th*=50**
Naive Algorithm	8.5(s)	13.17(s)	12.30(s)	14(s)	14.91(s)
Improved Algorithm	7.9(s)	10.1(s)	11.9(s)	12.9(s)	13(s)

In Table 2, our algorithm is faster than the naive one with 5 seconds less for execution time (*th*=42).

Table 2: Dataset 2: Size of raster =200×200, contains 1420, Interleave =73

	Threshold *th*=39	**Threshold *th*=40**	**Threshold *th*=41**	**Threshold *th*=42**	**Threshold *th*=50**
Naive Algorithm	44.57(s)	45.30(s)	47.80(s)	49.4(s)	53.76(s)
Improved Algorithm	40.5(s)	42.28(s)	43.04(s)	44.47(s)	50.39(s)

Table 3: Dataset 3: Size of raster =240×240, contains 1420, Interleave =73

	Threshold *th*=30	**Threshold *th*=40**	**Threshold *th*=42**	**Threshold *th*=50**	**Threshold *th*=70**
Naive Algorithm	36.81(s)	46.63(s)	48.53(s)	56.83(s)	67.03(s)
Improved Algorithm	32.57 (s)	44.57(s)	46.78(s)	53.55(s)	62.40(s)

As we can see in the table 3, our algorithm is still faster than the naive one. More precisely, our algorithm is always faster than the naïve one, whatever the value of the threshold. The user-defined threshold value has a direct impact on the performance of the improved algorithm.

5.2 The impact of the interleave size on the performance

The interleave value between the sorted rasters is important. It has also an impact on the performance of our algorithm. Choosing a low interleave implies sorting more rasters, which decreases the performance. In the other hand, choosing large interleave means sorting less rasters which is good for the performance, but in the same time, many rasters that are in the same interleave will not follow the same behavior as the sorted raster.
In Table 4 we show how the interleave size influences the performance of our algorithm on the data set 1. As an example, we have tested three interleave sizes. As we can see in table 4, the best performance is obtained by choosing the size 73. The choice of the interleave value depends on the nature of data and the frequency of its production.

Table 4: The impact of the interleave on the performance (Data set 1), Threshold =40

	Interleave =10	Interleave =20	Interleave =73
Naive Algorithm	13.17(s)	13.17(s)	13.17(s) (s)
Improved Algorithm	11.27(s)	10.53(s)	10.4

Our algorithm shows interesting potential, it should be improved by using other faster sorting algorithms and also using raster data sets with significant variation of data in the same raster.

6 Conclusion and discussion

In literature, different proposals have been implemented for raster processing, but no previous work had proposed our approach based on a value sort for a conditional selection of rasters. The processing proposed in this paper is rather specific and is in the field of the conditional selection, based on a maximal threshold, in a large raster data set. It corresponds to a concrete and useful operation for processing the set of rasters especially in the context of global climate change and agri-environment. Consequently, it was important to propose and develop new techniques to improve the computation.

We show in this paper, an idea for improved processing and the first associated tests. We have generated several data sets in order to analyze the influence of the different value properties on the process performance. We have to continue our tests to highlight in which cases our approach is suitable. For that, we have to analyze more precisely the impacts of the data, the interleave, the threshold, the number of selected rasters, etc. on the performance of the process. Our goal is to provide to computer scientists, information about the conditions in which the proposed improvement is interesting and should be used. Thanks to this information, we will produce recommendations about the different types of agri-environmental data suitable for this technique. Our proposal can also be directly applied on large vector of geo-referenced sensor data. We also plan to test our method on a real data set produced by sensor acquisition in real farms (soil and air moisture, temperature, etc.).

Here, we emphasize two main perspectives for the future improvement of our method:

1) The use of the General-Purpose logic on Graphics Processing Unit (GPGPU) technology can be experimented in order to provide a parallel processing of the computation. Different tests have been made for the optimization of operations on rasters using GPGPU by other authors [16-19]. Currently, there is no implementation proposal by GPGPU for our improved process. Our process can be massively parallelized, as several parts of the procedure are independent and can be run in parallel. For example, several rasters can be processed in parallel.
2) It is useless to pre-calculate and store the average of each raster in a persistent manner. The user chooses a sub-part of the raster (i.e., the geographical region of interest) and this sub-part is not known in advance. This region of interest can also be different for each new user query. Nevertheless, we could decompose each raster in several partitions (e.g., regular partitions) and pre-calculate and store persistently the average of the cell values for each partition. In this case, we could use these averages as

indicators to determine if our improved process should be used or not. For example, if the user-defined geographical region of interest is spatially included in a partition *p*, the system could decide to automatically run our improved method if the partition average (of the first raster in the sequence for example) is greater than the user-defined threshold. Different other parameters could be used to determine if it is useful to run our improved method.

To conclude, in our opinion, the research work presented in this paper opens the way to other interesting contributions that can be of interest in the field of agri-environment, especially for sensing data produced by remote sensing, agricultural internet of things and smart farming [20-26].

Acknowledgement

This work was funded by grants from the French program « investissement d'avenir » managed by the Agence Nationale de la Recherche of the French government (ANR), the European Commission (Auvergne FEDER funds) and « Région Auvergne » in the framework of LabEx IMobS 3 (ANR-10- LABX-16-01). We also acknowledge the support received from the Agence Nationale de la Recherche of the French government through the program « investissement d'avenir » 16-IDEX-0001 CAP 20-25 on the general topic of this paper.

References

1. Sawant, S., Durbha, S.S., Jagarlapudi, A.: Interoperable agro-meteorological observation and analysis platform for precision agriculture: A case study in citrus crop water requirement estimation. Computers and Electronics in Agriculture 138 (2017) 175-187
2. Lee, W.S., Ehsani, R.: Sensing systems for precision agriculture in Florida. Computers and Electronics in Agriculture 112 (2015) 2-9
3. Ahmedi, F., Ahmedi, L., O'Flynn, B., Kurti, A., Tahirsylaj, S., Bytyçi, E., Sejdiu, B., Salihu, A.: InWaterSense: An intelligent wireless sensor network for monitoring surface water quality to a river in Kosovo. International Journal of Agricultural and Environmental Information Systems 9 (2018) 39-61
4. Devadevan, V., Sankaranarayanan, S.: Forest fire information system using wireless sensor network. International Journal of Agricultural and Environmental Information Systems 8 (2017) 52-67
5. Kang, M.A., Pinet, F., Schneider, M., Chanet, J.P., Vigier, F.: How to design geographic database? Specific UML profile and spatial OCL applied to wireless Ad Hoc networks. 7th Conference on Geographic Information Science (AGILE'2004), Heraklion, GRC, April 29-May 1 2004 (2004) 289-299

6. Pinet, F.: Entity-relationship and object-oriented formalisms for modeling spatial environmental data. Environmental Modelling & Software 30 (2012) 80-91
7. Pinet, F., Miralles, A., Papajorgji, P.: Modeling : a central activity for designing and implementing flexible agricultural and environmental information systems. International Journal of Agricultural and Environmental Systems 1 (2010)
8. Machwitz, M., Hass, E., Junk, J., Udelhoven, T., Schlerf, M.: CropGIS – A web application for the spatial and temporal visualization of past, present and future crop biomass development. Computers and Electronics in Agriculture (2018)
9. Roussey, C., Bernard, S., Pinet, F., Reboud, X., Cellier, V., Sivadon, I., Simonneau, D., Bourigault, A.-L.: A methodology for the publication of agricultural alert bulletins as LOD. Computers and Electronics in Agriculture 142 (2017) 632-650
10. Ramar, K., Gurunathan, G.: Semantic web based agricultural information integration. International Journal of Agricultural and Environmental Information Systems 8 (2017) 39-51
11. Rabindra, B.: CloudGanga: Cloud computing based SDI model for ganga river basin management in India. International Journal of Agricultural and Environmental Information Systems 8 (2017) 54-71
12. Laurini, R., Thompson, D.: Fundamentals of spatial information systems (1992)
13. Kang, M.-A., Zaamoune, M., Pinet, F., Bimonte, S., Beaune, P.: Performance optimization of grid aggregation in spatial data warehouses. Int. J. Digital Earth 8 (2015) 970-988
14. Diamond, H. J., T. R. Karl, M. A. Palecki, C. B. Baker, J. E. Bell, R. D. Leeper, D. R. Easterling, J. H. Lawrimore, T. P. Meyers, M. R. Helfert, G. Goodge, Thorne P. W., 2013: U.S. Climate Reference Network after one decade of operations: status and assessment. Bull. Amer. Meteor. Soc., 94, 489-498. doi:10.1175/BAMS-D-12-00170.1.
15. Hozo, S.P., Djulbegovic, B., Hozo, I., 2005. Estimating the mean and variance from the median, range, and the size of a sample. BMC Medical Research Methodology 5, 13–13.https://doi.org/10.1186/1471-2288-5-13
16. Melesse, A.M., Weng, Q., Thenkabail, P.S., Senay, G.B.: Remote sensing sensors and applications in environmental resources mapping and modelling. Sensors 7 (2007) 3209-3241
17. Zhang, J., You, S., Gruenwald, L.: Parallel online spatial and temporal aggregations on multi-core CPUs and many-core GPUs. Information Systems 44 (2014) 134-154
18. Prasad, S.K., McDermott, M., Puri, S., Shah, D., Aghajarian, D., Shekhar, S., Zhou, X.: A vision for GPU-accelerated parallel computation on geo-spatial datasets. SIGSPATIAL Special 6 (2015) 19-26
19. Simion, B., Ray, S., Brown, A.D.: Speeding up Spatial Database Query Execution using GPUs. Procedia Computer Science 9 (2012) 1870-1879
20. data management and analytics in interdisciplinary research. Computers and Electronics in Agriculture 145 (2018) 130-141
21. Lehmann, R.J., Reiche, R., Schiefer, G.: Future internet and the agri-food sector: State-of-the-art in literature and research. Computers and Electronics in Agriculture 89 (2012) 158-174
22. O'Grady, M.J., O'Hare, G.M.P.: Modelling the smart farm. Information Processing in Agriculture 4 (2017) 179-187
23. Severino, G., D'Urso, G., Scarfato, M., Toraldo, G.: The IoT as a tool to combine the scheduling of the irrigation with the geostatistics of the soils. Future Generation Computer Systems 82 (2018) 268-273
24. Talavera, J.M., Tobón, L.E., Gómez, J.A., Culman, M.A., Aranda, J.M., Parra, D.T., Quiroz, L.A., Hoyos, A., Garreta, L.E.: Review of IoT applications in agro-industrial and environmental fields. Computers and Electronics in Agriculture 142 (2017) 283-297
25. Tzounis, A., Katsoulas, N., Bartzanas, T., Kittas, C.: Internet of Things in agriculture, recent advances and future challenges. Biosystems Engineering 164 (2017) 31-48

26. Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.-J.: Big Data in Smart Farming – A review. Agricultural Systems 153 (2017) 69-80