Towards a query language for spatiotemporal data based on a formal algebra

Carlos A. Romani[1], Gilberto Câmara[1], Gilberto R. Queiroz[1],
Karine R. Ferreira[1], Lubia Vinhas[1]

[1]Image Processing Department
INPE - National Institute for Space Research
- 12227-010, São José dos Campos – SP – Brazil

{carlos.romani, gilberto.camara, gilberto.queiroz}@inpe.br

{karine.ferreira, lubia.vinhas}@inpe.br

Abstract. *The monitoring of land use and cover changes is essential for understanding several environmental and socio-economic processes in the World. This monitoring requires novel software tools able to process big spatiotemporal data sets generated from Earth observation satellites efficiently. To generate land use and cover maps, researchers from different areas need high-level mechanisms to easily handle these big data sets. Thus, this paper presents a query language for spatiotemporal data based on an algebraic formalism. By defining and implementing a temporal interval algebra, we can answer questions about land use and cover changes. For each location in study area, a time series is extracted of pre-processed and classified Earth observation data.*

1. Introduction

To monitor constant changes in land use and cover over time there is a need map and classify geospatial data with periodic repetitions. Geospatial technology in earth observation grows exponentially, providing daily large data sets in high spatial and temporal resolution. When this information is organized sequentially in time and space it is created a coverage series from which time series can be extracted based on location, assisting in monitoring of land use and cover changes [Câmara et al. 2014]. Each change in a geographic location over time can be described as an event [Ferreira et al. 2014].

Currently the Geographic Information Systems (GIS) still does not have adequate tools to work with spatiotemporal data, being a challenge to geoinformatics researches the development of stable and efficient algorithms operating with large data sets [Câmara et al. 2016]. Other important topic is to improve the communication between user and software, favoring scientists from different fields of activity to use these resources with facility.

An algebra to represent temporal relationships between events is defined by [Allen 1990]. [Worboys 2005] proposes important concepts about spatiotemporal data and make a event-oriented approach. [Ferreira et al. 2014] proposes an algebra to spatiotemporal data, and define three data types as *time series, trajectory* and *coverage*. A *coverage* set at the same time-indexed location is defined as the *coverage series* or *raster time series*. With the rise of research in spatiotemporal events, the Allen's relationships were adapted for representing models of formal algebras mentioned in

[Maciel et al. 2017]. Formal algebra consists of the definition of types and operators on these types in high-level of abstraction, independent of programming language. These specifications help in the development of GIS applications. [Maciel et al. 2017] make an approach about events in land use change using big Earth observation data, defining a formalism to represent events in an interval temporal logic. [Bisceglia et al. 2012] proposes a query language for temporal SOLAP (Spatial On-Line Analytical Processing) called *TPiet-QL*, which supports land use data with a approach to the discrete changes in objects.

The objective of this paper is to propose a query language for spatiotemporal data based on formal algebra proposed by [Maciel et al. 2017], creating easy-to-use tools that perform complex tasks and return to the user important informations that can be used to analysis in land use and cover changes.

2. Methodology

The work is organized in three parts: the language definition in agreement with the predicates, the parser implementation, and the spatiotemporal data manipulation, operations and results presentation.

2.1. Language definition

Starting from the need a language for computational representation of spatiotemporal events, some operators were combined to form query expressions, based on questions about land use and cover changes in a given region. The predicate algebra is quoted in [Allen 1990, Ferreira et al. 2014, Maciel et al. 2017].

Questions related with an event occurred in a time interval are proposed by [Maciel et al. 2017] this way:

Which “Forest” areas have been turned into “Pasture” after the year of 2001?

$$\forall o \in O,\ \text{occur}(o, \text{“}Forest\text{”}, t1) \wedge \text{occur}(o, \text{“}Pasture\text{”}, t2) \wedge$$
$$next(t1,t2)\ where\ t1 = 2001,\ t2 = \{2002, ..., 2015\}$$

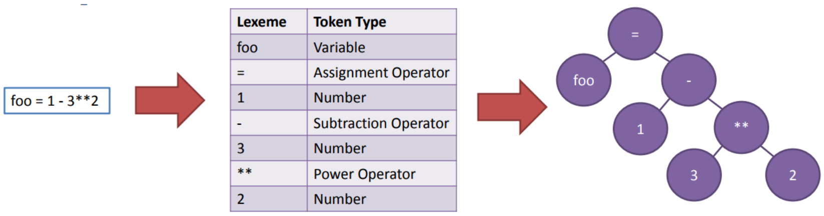
The operational algebra implemented is similar to the one shown above, with some changes in the predicates and syntax, creating composite expressions of a predicate, land use patterns and dates, depending on the operator. Below is shown the same expression, however in new proposed query language.

“meets(*Forest*,2001)&after(*Pasture*, 2001)”

This expression is a string interpreted by the parser, where each element is scanned and organized in hierarchical form as shown in Figure 1.

2.2. Parser

To implement this parser, we employed Flex (Fast Lexical Analyzer) and GNU Bison, which are two important tools for development of interpreters and compilers. Flex is a tool to assist in the development of lexical analyzer from the rule definition so that each character or character set. Lexical analyzer works as a scanner, by scanning all characters of the input string and translated to tokens, makes it possible to determine which characters belong to the alphabet of the language. Bison is a tool to develop a syntactic analyzer, or parser. The input of the parser are the tokens generated by the scanner, and the rules define the language syntax. The syntax is defined as a hierarchy of tokens and expressions which returns one result for each rule [Levine 2009]. These steps are represented in Figure 1.



Lexeme	Token Type
foo	Variable
=	Assignment Operator
1	Number
-	Subtraction Operator
3	Number
**	Power Operator
2	Number

Figure 1. Scanner and parser

Starting from grammatic of algebra to query language, must be defined the syntax rules of language, taking into account the precedence level of each element, building an output string with a determined standard. Each syntactic element can be an overload of rules which makes different output for same element.

Each *raster time series* can have different nomenclatures of patterns, so we have to provide the interpreter with the search expression and a set of metadata, associating each pattern name with the corresponding DN (Digital Number) in the images. These metadata are stored in memory to be used in the scan of the expression, comparing whether the input patterns have a corresponding and what their DN. The expression is parsed by the interpreter following the established syntax rules, especially when the expression is complex, with AND and OR operators. For each operator in the complex expression, the precedence level must be taken into account. Above we can see a sample of rules.

```
<ste> ::= <args> <coma> <expressions>
<expressions> ::= <expression> | <expressions> <op> <expression>
<expression> ::= before <lparen> <class> <coma> <date> <rparen>
   | after <lparen> <class> <coma> <date> <rparen>
   ...
<args> ::= <arg> | <args> <arg>
<arg> ::= <intnum> <coma> <class>
<class> ::= <string> | <name>
<date> ::= <year> - <month> - <day> | <year> - <month> | <year>
<year> ::= <intnum>
...
```

2.3. Raster Time Series

The organization and manipulation of spatiotemporal data is implemented in **R** [R Core Team 2016] and “rts” (Raster Time Series) package. **R** is a statistical environment of programming, using a high-level language and easily extensible to create and to use packages. The package “rts” depends of packages “raster” and “xts” (eXtensible Time Series), that aid in the raster files importation and date association.

In this work we used the clipping of an images classifications time series, with 30x30 pixels of dimension and 15 patterns from land use and cover from 2001 to 2016. The Figure 2 show the clip.

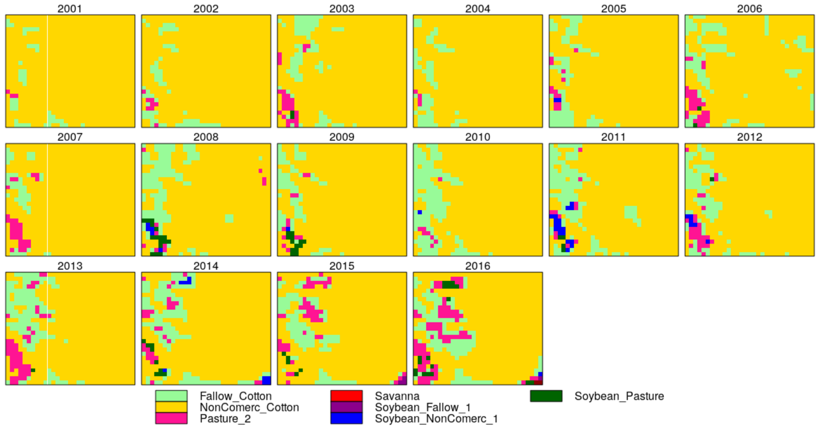


Figure 2. Clip of region

The *raster time series* is imported from an **R** object where the algorithms of extraction time series for each localization is obtained. Next stage is the parsing of the query expression. A **R** function call the parser algorithm, returning a set of logical operators involving temporal intervals and patterns of land use and cover. These logical comparisons will be included in **R** implementation, scanning each time series and returning a data cube with the same characteristics of input data, however containing only boolean values as answer of expression. The result is exported to files and can be used by any GIS software or manipulated with other post-process algorithm.

3. Results

The operators implemented are "before(*pattern, ti*)", "after(*pattern, ti*)", "meets(*pattern, ti*)", "meetby(*pattern, ti*)", "during(*pattern, ti, tj*)" and "equals(*pattern, ti, tj*)". The conjunction (AND) and disjunction (OR), can join two or more expressions. The data cube can be shown in **R** with a multiplot in sequence of images. The operation returns a binary data cube. The images are shown in black for true (1) and white for false (0). Figures 3 and 4 show results of some search expressions.

4. Conclusion

This study proposes a query language for spatiotemporal data using a formal algebra to describe events related with land use and cover. The main contribution of this work is to present an easer way of analyzing land use and cover changes through of spatiotemporal data, creating a new approach for a defined formalism based on events that occur over time. For data management we used the R programming language, that provide a large number of packages and techniques to data manipulation. The results of the experiments are in agreement with [Maciel et al. 2017], returns the answers to query expression based on temporal algebra. For later studies this method will be expanded to a greater number of functions and operators, also will be taken into account performance issues in big data sets. This query language can be integrated with a array database or part of a package to spatiotemporal analysis.

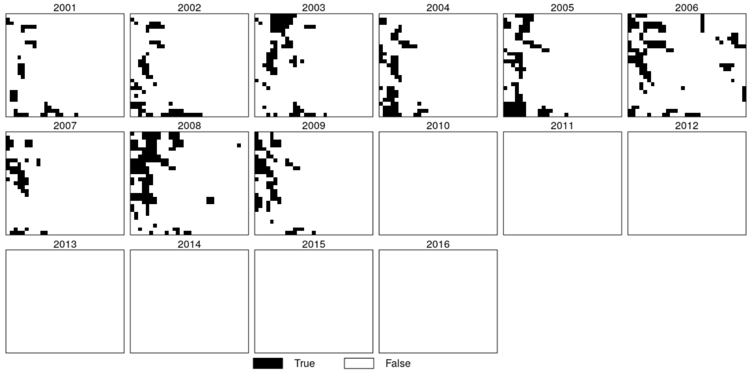


Figure 3. ***”before(Fallow_Cotton,2010)”***

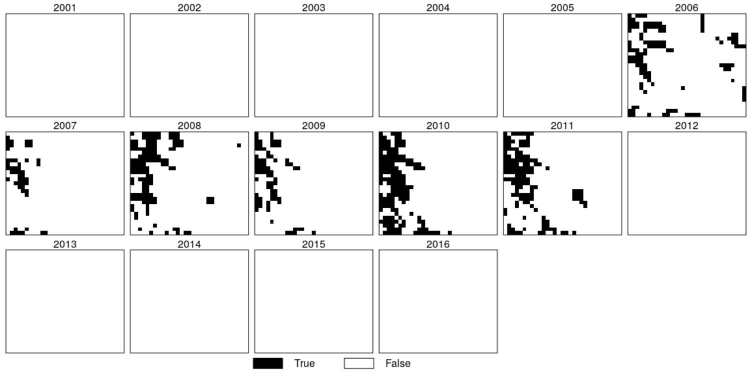


Figure 4. ***”meetby(Fallow_Cotton, 2006)&before(Fallow_Cotton, 2012)”***

References

Allen, J. F. (1990). Towards a General Theory of Action and Time. *Readings in Planning*, 23:464–479.

Bisceglia, P., Gómez, L., and Vaisman, A. (2012). Temporal SOLAP: Query language, implementation, and a use case. *CEUR Workshop Proceedings*, 866:102–113.

Câmara, G., Assis, L. F., Ribeiro, G., Ferreira, K. R., Llapa, E., and Vinhas, L. (2016). Big Earth Observation Data Analytics: Matching Requirements to System Architectures. *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial '16*, pages 1–6.

Câmara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., Jones, J., and Vinhas, L. (2014). Fields as a Generic Data Type for Big Spatial Data. *Geographic Information Science*, page in press.

Ferreira, K. R., Câmara, G., and Monteiro, A. M. V. (2014). *An Algebra for Spatiotemporal Data: From Observations to Events*. PhD thesis, National Institute for Space Research - INPE.

Levine, J. (2009). *Flex & Bison: Text Processing Tools*. O'Reilly Media.

Maciel, A. M., Vinhas, L., Camara, G., Maus, V., and Assis, L. F. F. G. (2017). STILF - A spatiotemporal interval logic formalism for reasoning about events in remote sensing data. Number February, pages 4558–4565.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Worboys, M. (2005). Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19(1):1–28.