Brazilian Journal of Cartography (2017), Nº 69/5, Special Issue GEOINFO 2017: 981-989
Brazilian Society of Cartography, Geodesy, Photgrammetry and Remote Sense
ISSN: 1808-0936

# POSTGIS-T: TOWARD A SPATIOTEMPORAL *POSTGRESQL* DATABASE EXTENSION

*PostGIS-T: Rumo a uma Extensão Espaço-Temporal do Banco de Dados PostgreSQL*

**Rolf E. O. Simoes, Gilberto R. de Queiroz, Karine R. Ferreira, Lubia Vinhas & Gilberto Camara**

**Instituto Nacional de Pesquisas Espaciais - INPE**
**Divisão de Processamento de Imagens - DPI**
Av. dos Astronautas, 1758, 12227-010 São José dos Campos, SP, Brasil
{lubia.vinhas, gilberto.queiroz, karine.ferreira, gilberto.camara}@inpe.br

***Received on January 8, 2017/ Accepted on April 27, 2017***
***Recebido em 8 de Janeiro, 2017/ Aceito em 27 de Abril, 2017***

## ABSTRACT

The temporal dimension of spatial data has been the subject of discussion in the literature for a long time. While there are numerous Database Management System (DBMS) solutions for spatial dimension, we do not observe the same situation for spatiotemporal data. Considering this gap, our purpose is to design and implement an extension to PostgreSQL DBMS that is based on a formal spatiotemporal algebra in order to incorporate representations of spatiotemporal data within the DBMS. The proposed extension can be used in a large range of applications. We intend this extension to be a reasonable framework to store and handle observational remote sensing data usually present in applications like animal migration researches, wildfires monitoring, vessel tracking for monitoring fishing, and other similar applications. In this work, we show how to apply it in a case study based on spatiotemporal data collected from drifting buoys belonging to the NOOA's Global Drifter Program.

**Keywords:** Database, Spatiotemporal, Extension, PostGIS-T.

## RESUMO

A dimensão temporal dos dados espaciais tem sido objeto de discussão na literatura há muito tempo. Enquanto existem inúmeras soluções para Sistemas de Gerenciamento de Banco de Dados (SGBD) para a dimensão espacial, não observamos a mesma situação para os dados espaço-temporais. Considerando esta lacuna, nosso objetivo é projetar e implementar uma extensão para o *PostgreSQL* que se baseia em uma álgebra espaciotemporal formal, a fim de incorporar representações de dados espaciotemporais dentro do SGBD. A extensão proposta pode ser usada em uma ampla gama de aplicações. Pretendemos que esta extensão seja uma estrutura razoável para armazenar e manusear dados observacionais de sensoriamento remoto normalmente presentes em aplicações como pesquisas de migração de animais, monitoramento de incêndios, monitoramento de embarcações navais de pesca e similares. Neste trabalho, mostramos como aplicá-lo em um estudo de caso baseado em dados espaço-temporais coletados a partir de boias à derivada pertencentes ao programa *Global Drifter* da NOOA.

**Palavras-chave**: Banco de Dados, Espaço-Temporal, Extensão, PostGIS-T.

## 1. INTRODUCTION

Earth Observation data generation has been increasing in the last decades. As this phenomenon occurs a great amount of data are daily collected by different missions such as *CBERS* in Brazil/China, *Landsat* in the USA and *Sentinel* in Europe. The development of mobile positioning technologies and its low costs are also factors that enable spatial data gathering through time.

These different data sources are associated with temporal dimension, and allow the monitoring of spatially located objects in time, and the time analysis by increasing the temporal resolution of the observations. The collection, representation and processing of this data have been largely facilitated by database management systems (DBMS) and their spatial extensions, which are based on international standards such as the OGC Simple Feature Specification (HERRING, 2011, OGC, 2010, OGC, 2011) and ISO geographic information standards (KRESSE and FADAIE, 2004). Furthermore, while there are numerous DBMS solutions supporting the spatial dimension, we do not observe the same situation for spatiotemporal data.

The temporal dimension of spatial data has been the subject of discussion in the literature for a long time. Additionally, some conceptual systems regarding to representation of spatiotemporal data have been proposed (CAMARA *et al.*, 2014; FERREIRA *et al.*, 2014; ERWIG *et al.*, 1999). One of these systems, particularly discussed in Ferreira *et al.* (2014), is structured around the concept of observations, the basic unit of data acquisition of a spatiotemporal phenomenon. From observations, it is possible to generate three types of spatiotemporal data: *time series*, *trajectories* and *coverages*. With these three types, it is possible to represent the geo-ontological concepts of object and field and to define a space-time algebra. The authors have implemented their work as a C++ library, which works as middleware between the actual sources of data (databases of flat files, for example) and their conceptual model.

We have observed a number of solutions for processing spatiotemporal data as client side solutions. This approach presents as disadvantages the introduction of an overhead by transferring data between processes or even between machines when those data are managed over a network. Differently from that, we propose a model that works inside the database system.

Besides that, this approach can avoid memory issues for huge volume of data since this is a subject solved in the context of relational DBMS such as PostgreSQL. Moreover, spatiotemporal data usually consumes a high volume of memory space. Considering this drawback, our goal is to implement an extension to the PostgreSQL DBMS to provide support for spatiotemporal types within the DBMS. The PostGIS *geometry* type is used as a basic spatial representation for the extension. Furthermore, temporal dimension was integrated to build our spatiotemporal type.

Here, we propose a spatiotemporal database extension. Based on the work of Ferreira *et al.* (2014), we introduce three new spatiotemporal data types for the DBMS PostgreSQL. Moreover, we implemented some algebra functions for those data. In order to demonstrate how our extension can manipulate real data observations, we conducted a case study based on the Global Drifter Program (GDP) database.

This paper is based on Simoes *et al.* (2016), previously presented in XVII Brazilian Symposium on Geoinformatics (GeoInfo 2016) (http://www.geoinfo.info/geoinfo2016/).

## 2. BACKGROUND

According to Sinton (1978), space, time and theme (or a quantity measure) are the three dimensions of geographical structure and observation. In such way, it is possible to observe by fixing one dimension, controlling another and measuring the other. Hence, six types or structure of observation can be produced. Proceeding in this manner, Ferreira *et al.* (2014) claim that we can capture all kind of spatiotemporal phenomena exhaustively with three of them:

**time series**: fix space, control time and measure theme;

**trajectory**: fix theme, control time and measure space;

**coverage**: fix time, control space and measure theme;

Time series and trajectory play an important role considering the necessity to analyze data along time. The main difference between them remains in space dimension: in the case of trajectory we are interested in measuring the space locations of an given observed phenomena, whereas in time series the measured data is gathered at a fixed space.

It is easy to see that the fix operation defines a domain from which the measured data must be filtered in. In this manner, a geo-located time series are sequences of observations over time of a measured phenomenon that takes place in a given space domain. For instance, Landsat series spans over 40 years and is a very informative temporal record of radiance of geo-located sites represented by the pixels (ROY *et al.*, 2014). This database can provide significant data about each pixel spectral response over a time series that can be further analyzed to give us information about land use and land cover change (MAUS, 2016).

On the other hand, trajectories can be seen as a sequence in time dimension of geo-located observed geometries (points, lines, polygons or volumes) that are associated to a given theme (e.g. objects). For instance, monitoring oceanic buoys give us a rich data about their positions, water temperature and salinity. One can be interested in tracking buoys positions over time to capture oceanic surface chains (LUMPKIN and PAZOS, 2007).

Here, the time dimension organizes different space locations of a fixed theme (the drifter). In another example, the policy to prevent Dengue epidemic may depend on monitoring egg traps over time (REGIS *et al.*, 2009). In this case, one may be interested to know all locations over time that presents a given high number of captured mosquito eggs (threshold theme). The main difference between these two examples of trajectories is the kind of fixed theme. In the first one, the fixed theme was an identifiable object entity, a drifter. In the second one, we used a measured data not related to a specific object entity but a condition that may involve a set of objects. This condition refers to object properties and can denote events. For example, if we are interested in flood monitoring in some urban area, we may relate the event *flood* to a condition of water precipitation metered by a set of pluviometers.

Objects and events play a key role on the interpretation of a spatiotemporal data (FERREIRA *et al.*, 2014; WORBOYS, 2005). In this approach, an object is any identifiable entity over time and an event is an episode on time that may relate to one or more objects. Episodes have a definite begin and end. Events may be distinguished by punctual occurrence, if it occurs instantaneously, or durative, if it takes some time (GALTON, 2004). An event does not change over time and we can derive them from conditions of spatial and non-spatial properties of objects, as we can see from the example above. Figure 1 summarizes this model.

Spatiotemporal structures can be implemented on computer systems as data types. A data type is a set of values over which we can define some operations. To formalize these ideas, Ferreira *et al.* (2014) follow Guttag and Horning (1978) and propose an algebraic specification that consists of (1) definitions of type names, their domains, ranges, and operations; (2) a set of axioms that expresses truth relations among those operations. PostgreSQL works similarly: all data has a type with an explicit or implicit ranges; all functions operates over some types and returning new data. To comply with Ferreira *et al.* (2014) abstract model, we just need to guarantee PostGIS-T functions to be implemented in such a way that its behavior is consistent with axioms specification. In what follows, we describe how our extension implements and represents spatiotemporal data.

## 3. POSTGIS-T MODEL

PostGIS-T introduces the `SPATIOTEMPORAL` type, a composite PostgreSQL type that stores spatiotemporal data and metadata. The main difference between this type and those presented in Ferreira *et al.* (2014) is that `SPATIOTEMPORAL` type can represent both *time series*, *trajectories* and, *coverages*. In this regard, the prototype does not work with the idea of abstract type that is specialized later.

In order to access and manipulate the data, we must define functions. We have implemented each operator declared in Ferreira *et al.* (2014) model definition. We list all functions signature in Table 1.

PostGIS-T stores observations as tuples in one or more relations that can be further queried to instantiate `SPATIOTEMPORAL` data. To get a new `SPATIOTEMPORAL` data, we use `TST_SPATIOTEMPORAL()` aggregate function.

PostGIS-T was built over PostgreSQL and PostGIS. For example, spatial representations are `GEOMETRY` values, a type introduced by PostGIS. As a first prototype we have limited measure values as `NUMERIC` type, so that an observation must be a triple (`TIMESTAMP`, `GEOMETRY`, `NUMERIC`). In order to instantiate `SPATIOTEMPORAL` data we must call `TST_SPATIOTEMPORAL()` inside a query informing where to find these values.

Spatiotemporal data may be interpreted differently depending on the underlying phenomena it represents. As previously discussed, we could conceive an observation as taking place continuously or instantaneously. Our data type was defined to accommodate all spatiotemporal data phenomena without a specific semantic. For example, we do not know in advance if we should conceive an observation as an *occurrent* or a *durative* one, or if the sample observation is a time series or a trajectory. To overcome this limitation, we have implemented some functions that can be combined in order to handle the right phenomena interpretation. These functions are `TST_ESTIMATE_MEASURE()`, `TST_ESTIMATE_LOCATION()`, `TST_RESAMPLE_TIME()`, and `TST_COVERAGE()`.

If we are interested in how to get an approximate measure between two empirical observed phenomenon we must take into account its underlying nature, that is, if it refers to an object or to an event. For example, we took observations of a drifter floating on the ocean at times 10:00AM and 11:00AM, and we are interested to know its location at 10:30AM. We may assume that a linear interpolation would give us a good approximation and so we call `TST_ESTIMATE_LOCATION()` function informing the `SPATIOTEMPORAL` data, the time of interest to calculate the interpolation (in this case 10:30AM), and the interpolation method name `'LINEAR'` as parameters. For now, we have implemented a small set of interpolation methods that we can use, `'LINEAR'`, `'LAST'` or `'NEAREST'`, meaning, respectively, simple linear interpolation, last registered location or measure before or equal a given time, and the closest time registered location or measure of a given time. The process to estimate a measure from a `SPATIOTEMPORAL` is analogous.

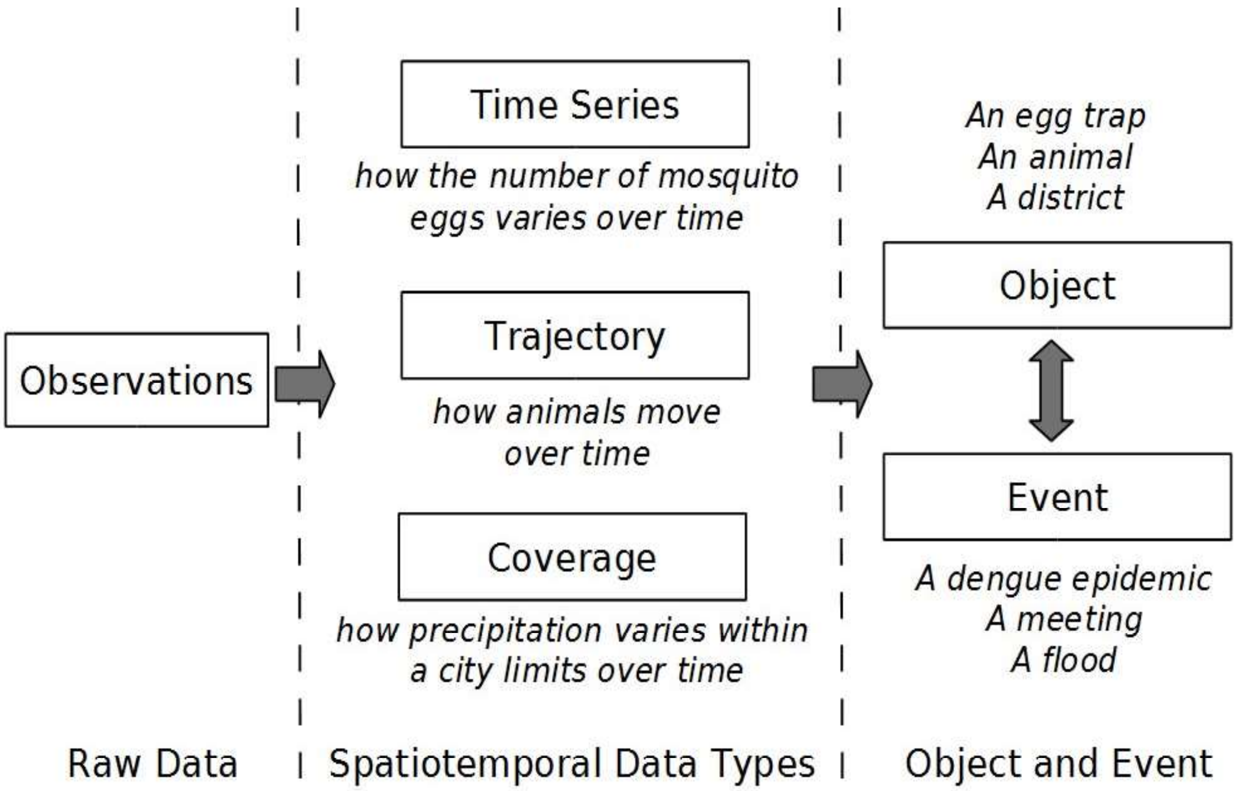


Fig. 1 - The proposed model. Source: Ferreira *et al.* (2014).

Table 1: List of all defined PostGIS-T functions

| Function signature | Return type |
|---|---|
| TST_AFTER(SPATIOTEMPORAL, TIMESTAMP) | SPATIOTEMPORAL |
| TST_BEFORE(SPATIOTEMPORAL, TIMESTAMP) | SPATIOTEMPORAL |
| TST_BEGINS(SPATIOTEMPORAL), TST_ENDS(SPATIOTEMPORAL) | TIMESTAMP |
| TST_BETWEEN(SPATIOTEMPORAL, NUMRANGE) | SPATIOTEMPORAL |
| TST_COVERAGE(SPATIOTEMPORAL, INTEGER, INTEGER, TEXT) | SPATIOTEMPORAL |
| TST_DIFFERENCE(SPATIOTEMPORAL, GEOMETRY) | SPATIOTEMPORAL |
| TST_DURING(SPATIOTEMPORAL, TSRANGE) | SPATIOTEMPORAL |
| TST_EQUALS(SPATIOTEMPORAL, SPATIOTEMPORAL) | BOOLEAN |
| TST_ESTIMATE_LOCATION(SPATIOTEMPORAL, TIMESTAMP, TEXT) | GEOMETRY |
| TST_ESTIMATE_MEASURE(SPATIOTEMPORAL, TIMESTAMP, TEXT) | NUMERIC |
| TST_GREATER(SPATIOTEMPORAL, NUMERIC) | SPATIOTEMPORAL |
| TST_HULL(SPATIOTEMPORAL) | GEOMETRY |
| TST_INTERPOLATOR(SPATIOTEMPORAL) | TEXT |
| TST_INTERSECTION(SPATIOTEMPORAL, GEOMETRY) | SPATIOTEMPORAL |
| TST_LESS(SPATIOTEMPORAL, NUMERIC) | SPATIOTEMPORAL |
| TST_LOCATION(SPATIOTEMPORAL, TIMESTAMP) | GEOMETRY |
| TST_MEASURE(SPATIOTEMPORAL, GEOMETRY, TEXT) | NUMERIC |
| TST_MEASURE(SPATIOTEMPORAL, TIMESTAMP) | NUMERIC |
| TST_MIN(SPATIOTEMPORAL), TST_MAX(SPATIOTEMPORAL) | NUMERIC |
| TST_OBSERVATIONS(SPATIOTEMPORAL) | INTEGER |
| TST_RESAMPLE_TIME(SPATIOTEMPORAL, TSRANGE, INTEGER, TEXT) | SPATIOTEMPORAL |
| TST_SETINTERPOLATOR(SPATIOTEMPORAL, TEXT) | SPATIOTEMPORAL |
| TST_SPATIOTEMPORAL(TIMESTAMP, GEOMETRY, NUMERIC) | SPATIOTEMPORAL |

Other useful application of interpolators is re-sampling data. PostGIS-T is able to re-sample time observations between a time range at a regular time resolution with the function `TST_RESAMPLE_TIME()`. This function returns a regular time spaced `SPATIOTEMPORAL` data whose locations and measures were estimated according to a given interpolation method (see more details in section 4).

Furthermore, if we are interested in re-sampling our observations through space in order to produce what Ferreira *et al.* (2014) calls *coverage*, we use the function `TST_COVERAGE()`. This function returns a `SPATIOTEMPORAL` data where observations refer to a regular extent over which measures are aggregated into an unique value according to a given aggregate strategy (e.g. `'COUNT'`, `'AVG'`, `'MIN'`, `'MAX'`, and `'AREA'`). The result is a time flattened spatiotemporal data where no gap and no overlapping area exists between two adjacent extents. In Ferreira *et al.* (2014) model, this represents a *Coverage*.

Other functions are related to operations that retrieve `SPATIOTEMPORAL` properties or subsets of the spatiotemporal data. For instance, the functions `TST_BEGINS()` and `TST_ENDS()` indicate the start and the end times for sampling, whereas `TST_HULL()` gives the convex hull polygon where observations took place. Finally, to get the maximum and the minimum values of the measured data, we must use the functions

`TST_MIN()` and `TST_MAX()`, respectively. On other hand, the `TST_DURING()` function returns all observations that have been made at a given time range. Likewise, to get only a given location samples, we use the function `ST_INTERSECTION()` passing to it the area or point of interest.

In the following section, we demonstrate an application of how to use PostGIS-T.

## 4. THE GLOBAL DRIFTER PROGRAM: A CASE STUDY WITH REAL SPATIOTEMPORAL DATA

Regarding spatiotemporal data and its complexity, we chose the satellite-tracked surface drifting buoy (drifter) data to evaluate and get experienced with the extension implementation details in the PostgreSQL environment. The Global Drifter Program (GDP) is a branch of the National Oceanic and Atmospheric Administration (NOAA). It aims to maintain a global satellite-tracked sea surface drifting buoys and to provide the data set for scientific purposes, such as climate forecast and climate research and monitoring. In this manner, GDP produces observations from most areas of the world oceans at sufficient density to map the mean currents at one degree resolution (LUMPKIN and PAZOS, 2007).

Drifter is a surface buoy connected with a subsurface drogue. Its observations have been largely used in oceanographic and climate researches. The main use of this data is to map oceanic surface currents of different seas and oceanic regions of the planet. Also, the data can be used to calibrate satellite sensors. Each drifter has an unique identifier code and is equipped with sensors that periodically measure properties such as water salinity and surface temperature. All these data are subsequently transmitted to satellites. The drifter position and velocity are usually inferred by *Doppler shift*, which occurs during the transmission step. The positioning system, also known as *Argos*, provides drifter locations with *O*(100m) errors. The raw data are then assembled and normalized by the Drifter Data Assembly Center of the Atlantic Oceanographic and Meteorological Laboratory (DAC/AOML).

The used GDP drifter database contains 2,263,842 locations collected with their respective zonal and meridional velocities observations for 408 drifters worldwide. The time resolution of each observation is one hour. More information about how this database was collected can be seen in Elipot *et al.* (2016).

All data samples were loaded in a table of observations as defined in the Code 1. Subsequently, we proceeded with data instantiation by calling the aggregate function `TST_SPATIOTEMPORAL()`. All the following queries use this table.

```
CREATE TABLE buoy_obs_st (
  buoy_id    INTEGER PRIMARY KEY,
  spatiotemp SPATIOTEMPORAL
);
```

Code 1 - Definition of the spatiotemporal table `buoy_obs_st`.

Sometimes it is useful to change the amount of observations of a given data. In PostGIS-T, we can obtain a new sample of a trajectory by using the `TST_RESAMPLE_TIME()` function. This function receives as parameter the spatiotemporal data, the time interval that we are want re-sampling, the number of observations to be re-sampled, and the interpolation method. Note that the extension makes no assumption about the observation continuity or duration. In this regard, an appropriate interpolation method must be informed by the user. The queries in Code 2 shows how do we re-sample observations. A graphical result is presented in Figure 2.

```
SELECT buoy_id,
       TST_RESAMPLE_TIME(
           spatiotemp,
           TST_OBSERVATIONS(
               spatiotemp) * 2,
           'LINEAR')
FROM buoy_obs_st;
```

```
SELECT buoy_id,
       TST_RESAMPLE_TIME(
           spatiotemp,
           TST_OBSERVATIONS(
               spatiotemp) / 10,
           'LINEAR')
  FROM buoy_obs_st;
```

Code 2 - Re-sampling on time. Query on the top (bottom) reduces (increase) the time resolution.

Re-sampling technique may be employed to synchronize data observations, in order to speed-up queries that performs sequential time calculations. Comparing trajectories (a) and (b) from the Figure 2, we can note that some trajectory sections can be well approximated by few interpolated observations. These sections mainly resemble straight segments.

Another useful application of these data is to estimate the mean velocity of ocean currents over different regions and at a given period of the year. This is a very important question related to climate research. Velocity may be represented by a vector that informs us about the direction and speed magnitude of the moving entity. Here, re-sampling technique may be useful if we would like to measure the mean direction of currents.

Suppose now that we are interested in measuring the mean velocity magnitude of a drifter in a small time range (for the sake of simplicity). How can we proceed in PostGIS-T? First we need a function that returns the observations of a given time interval. This function is `TST_DURING()` which returns a `SPATIOTEMPORAL` data. From this result, `TST_COVERAGE()` creates a regular grid whose extent is the same as the observations location bounding box.

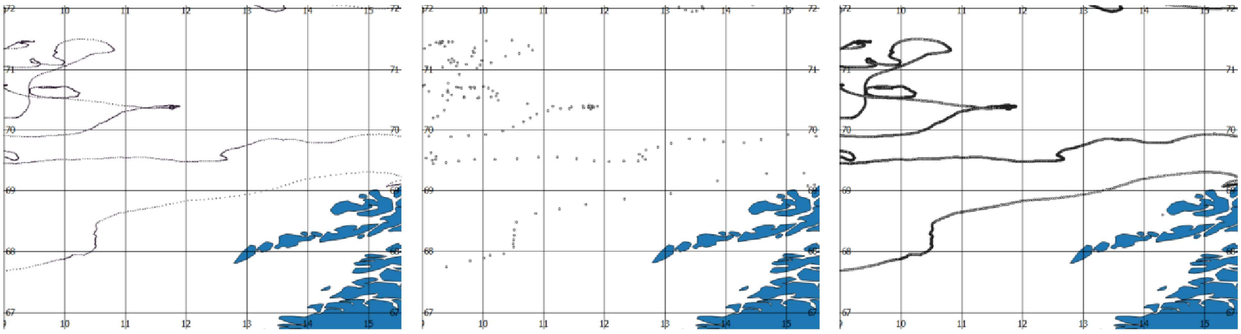

Fig. 2 - Trajectory re-sampling. From left to right: original data, re-sampling for every 10 hours, re-sampling for every 30 minutes. Geometries generated in PostGIS-T extension. Images generated in QGIS software.

From Figure 3, we can see that not all cells grid contains a drifter observation. Only those regions that have at least one register has a measure value equals to that of a velocity average. In order to fill those grid cells we would need a spatial interpolator. The process to get a coverage of velocity magnitudes is similar.

```
SELECT buoy_id,
       TST_COVERAGE(
           TST_DURING(
               spatiotemp,
               TSRANGE(
         '2015-05-18 14:00:00',
         '2015-05-20 14:00:00', '[]')
           ), 7, 13, 'AVG')
  FROM buoy_obs_st
 WHERE buoy_id = 132470;
```

Code 3 - PostGIS-T coverage generation SQL code.

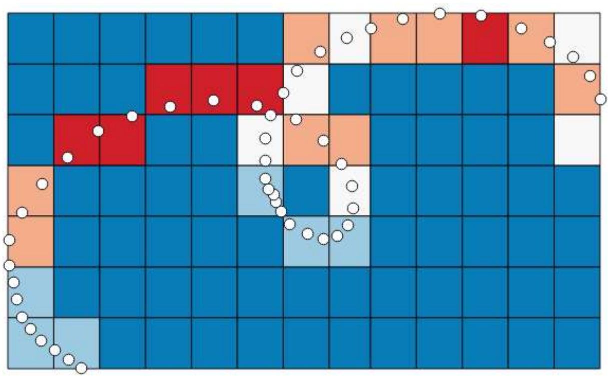

Fig. 3 - Coverage calculated from drifter mean velocity. Blue-Red colors denotes lower-higher velocities. Geometries generated in PostGIS-T extension. Image generated in QGIS software.

## 5. CONCLUSIONS

This paper addressed the problem of how spatiotemporal data can be represented, stored and processed. We proposed a database extension that allows the practitioners of the community to manipulate such datasets on the database side.

Our extension was based on conceptual model of Ferreira *et al.* (2014) and, as a starting point, we implemented it with some adaptations to comply with relational database environment. For example, in this preliminary version we did not provide a way to extend the base of interpolation methods used as a parameter of functions like `TST_RESAMPLE_TIME()`. In spite of that, the implementation shows us that the spatiotemporal model proposed in Ferreira *et al.* (2014) is feasible in an relational DBMS context.

Our first approach suggests that this model may be indicated more to applications of sparse geo-referenced data like movable objects (e.g. drifters, ship trajectories) and observed events (e.g. wildfires, disease occurrences). These applications should work better over snapshots of the original data as the task of packing a huge spatiotemporal tuples is expensive. However, it is too early to observe some processing improvement from our data type columnar design.

Further research and development consist of: (a) designing a compact and efficient disk storage layout for values of `SPATIOTEMPORAL` type; (b) introduce the notion of subtypes of `SPATIOTEMPORAL` as type modifiers (`TIMESERIES`, `TRAJECTORY` and `COVERAGE`), which will give more constraint about the data and the result of operations; (c) how to include spatiotemporal indexes that could take advantage of approximations of spatiotemporal data; (d) explore the extension with *big earth observation data* applications in a database cluster environment.

## ACKNOWLEDGEMENTS

This work was partially supported by FAPESP, the São Paulo State Research Foundation, under grant 2014-08398-6 (FAPESP e-science program) and by CNPq, the Brazilian National Council of Scientific and Technological Development, through PhD. grant 140684/2016-6 (Rolf Simoes); Gilberto Camara is also supported by CNPq (grant 3121512014-4).

## REFERENCES

Camara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., Jones, J., and Vinhas, L. **Fields as a generic data type for big spatial data**. In: International Conference on Geographic Information Science, Springer. p. 159–172, 2014.

ELIPOT, S., LUMPKIN, R., PEREZ, R. C., LILLY, J. M., EARLY, J. J., and SYKULSKI, A. M. A global surface drifter data set at hourly resolution. **Journal of Geophysical Research: Oceans**, v.121, n.5, p.2937–2966, 2016.

ERWIG, M., GU, R. H., SCHNEIDER, M., VAZIRGIANNIS, M., Spatio-temporal data types: An approach to modeling and querying moving objects in databases. **GeoInformatica**, v.3, n. 3, p.269–296, 1999.

FERREIRA, K. R., CAMARA, G., and MONTEIRO, A.M.V. An algebra for spatiotemporal data: From observations to events. **Transactions in GIS**, v.18, n.2, p. 253–269, 2014.

GALTON, A. Fields and objects in space, time, and space-time. **Spatial cognition and computation**, v.4, n.1, p.39–68, 2004.

GUTTAG, J. V. and HORNING, J. J. The algebraic specification of abstract data types. **Acta informatica**, v.10, n.1, p.27–52, 1978.

HERRING, J. OpenGIS implementation standard for geographic information - simple feature access - Part1: Common architecture. **OGC Document**, v.21, n.4, p.122–127, 2011.

KRESSE, W. and FADAIE, K. **ISO standards for geographic information**. Springer Science & Business Media, 2004.322p.

LUMPKIN, R. and PAZOS, M. **Measuring surface currents with surface velocity program drifters: the instrument, its data, and some recent results**. In: GRIFFA A., KIRWAN Jr. A. D., MARIANO, A. J., ÖZGÖKMEN, T. and ROSSBY, H. T., editors, Lagrangian analysis and prediction of coastal and ocean dynamics. Cambridge University Press New York, NY, chapter 2, p. 39–67, 2007.

MAUS, V. W. **Land Use and Land Cover Monitoring Using Remote Sensing Image Time Series**. PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2016. 94p.

OGC - Open Geospatial Consortium. **OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture**. Technical report, Open Geospatial Consortium, 2010. http://www.opengeospatial.org/standards/sfa

OGC - Open Geospatial Consortium. **OpenGIS Implementation Standard for Geographic information – Simple feature access - Part 2: SQL option**. Technical report, Open Geospatial Consortium, 2011. http://www.opengeospatial.org/standards/sfs

REGIS, L., SOUZA, W. V., FURTADO, A. F., FONSECA, C. D., SILVEIRA Jr, J. C., RIBEIRO Jr, P. J., MELO-SANTOS, M. A. V., CARVALHO, M. S., and MONTEIRO, A. **An entomological surveillance system based on open spatial information for participative dengue control**. In: Anais da Academia Brasileira de Ciências, v.81, n.4, p.655–662, 2009.

ROY, D. P., WULDER, M. A., LOVELAND, T. R., WOODCOCK, C. E., ALLEN, R. G., ANDERSON, M. C., HELDER, D., IRONS, J.R., JOHNSON, D.M., KENNEDY, R., SCAMBOS, T.A., SCHAAF, C.B., SCHOTT, J.R., SHENG, Y., VERMOTE, E.F., BELWARD, A.S., BINDSCHADLER, R., COHEN, W.B., GAO, F., HIPPLE, J.D., HOSTERT, P., HUNTINGTON, J., JUSTICE, C.O., KILIC, A., KOVALSKYY, V., LEE, Z.P., LYMBURNER, L., MASEK, J.G., McCORKEL, J., SHUAI, Y., TREZZA, R., VOGELMANN, J., WYNNE, R.H., ZHUD, Z. (2014). Landsat-8: Science and product vision for terrestrial global change research. **Remote Sensing of Environment**, v.145, p.154-172.

SINTON, D. **The inherent structure of information as a constraint to analysis: mapped thematic data as a case study**. In: DUTTON, G., editor, Harvard Papers on Geographic Information Systems, Harvard University Cambridge, MA., v.6, p.1–17, 1978

SIMOES, R., QUEIROZ, G., FERREIRA, K., VINHAS, L., and CAMARA, G. **Postgis-T: toward a spatiotemporal postgresql database extension**. In: Proceedings of XVII Brazilian Symposium on Geoinformatics (GeoInfo), 2016, Campos do Jordão, Brazil.

WORBOYS, M. Event-oriented approaches to geographic phenomena. **International Journal of Geographical Information Science**, v.19, n. 1, p.1–28, 2005.