

Armazenamento e representação de dados espaço-temporais: desenvolvimento e aplicações

Lorena A. Santos, Jéssica D. Santos, Rolf E. O. Simoes

14 de setembro de 2016

Sumário

1	Introdução	4
2	Revisão Teórica	6
2.1	Diferentes visões sobre o mesmo conjunto de observação	6
3	Manipulação e Análise de Dados Espaço-Temporal no R	9
3.1	Estudo de caso	11
4	Modelo PostGIS-T	18
4.1	Global Drifter Program: um exemplo de aplicação	19
5	Conclusão	22

Resumo

Os dados espaço-temporais têm sido por muito tempo objeto de discussão na literatura. Enquanto existem inúmeras soluções para dados espaciais em Sistemas Gerenciadores de Banco de Dados, não observamos a mesma situação para os dados *espaço-temporais*. Considerando essa lacuna, o objetivo do presente trabalho é propor duas diferentes extensões que incorporam representações espaço-temporais, uma em ambiente R e outra no SGBD PostgreSQL.

1 Introdução

A geração de dados de observação da Terra tem crescido nas últimas décadas. O desenvolvimento e o barateamento de novas tecnologias móveis capazes de coletar dados geoespaciais tem possibilitado o seu uso massivo em uma diversidade de aplicações.

Esses dados de diferentes fontes estão cada vez mais associados à dimensão temporal, seja por permitirem o monitoramento no tempo de objetos espacialmente localizados, seja por possibilitarem a realização de análises no tempo pelo aumento da resolução temporal das observações. O armazenamento, a representação e o processamento desse volume de dados tem sido largamente facilitados pelos sistemas gerenciadores de bancos de dados (SGBD) e suas extensões espaciais.

A dimensão temporal dos dados geográficos tem sido objeto de discussão na literatura. Muitas propostas de modelos de dados espaço-temporal tem sido apresentadas, abordando a evolução de objetos e campos ao longo do tempo. Algumas são específicas para mudanças discretas em objetos [23][12], outras para objetos em movimento [10] [13] e ainda outros para campos [15] [17] [3]. A demanda por modelos que descrevem eventos tem incentivado a recente pesquisa sobre modelagem de dados espaço-temporal [24] [8].

Em [5], uma abordagem para se obter eventos a partir do conceito de observações é proposta. Os eventos são representados de forma explícita, variando objeto e campos ao longo do tempo. Um evento contém informações dos objetos envolvidos e sobre quando e onde ocorreu. Nessa abordagem, a observação é a unidade básica de aquisição de um fenômeno espaço-temporal.

Tomando como base os diferentes tipos de dados espaço-temporal proposto por [5], iremos apresentar neste trabalho, duas propostas de implementação destes tipos de dados em diferentes ferramentas. Primeiro, apresentaremos o modelo utilizando o ambiente R e um segundo sistema que funciona dentro do sistema gerenciador de banco de dados PostgreSQL.

A primeira proposta é carregar os dados de observação para o ambiente R, sendo essa uma ferramenta de software amplamente utilizada para a análise de dados [21]. Ele fornece uma ampla variedade de métodos estatísticos (análise de séries temporais, classificação e agrupamento) e um ambiente de programação de alto nível e linguagem adequada para rápido desenvolvimento de novos algoritmos. R é estendida através de pacotes. Embora existam muitos pacotes para a manipulação de dados espaciais e análise, alguns deles podem lidar adequadamente com a dimensão temporal dos dados espaciais.

Para a segunda proposta, observamos que na literatura existe uma série de soluções para o processamento de dados espaço-temporais como soluções do lado do cliente. Esta abordagem apresenta como desvantagem a introdução de uma sobrecarga de transferência de dados entre processos ou mesmo entre máquinas, quando esses dados são gerenciados em um ambiente de rede. Tendo em vista essa lacuna, nossa proposta é implementar uma extensão para o SGBD Postgres/PostGIS de modo a incorporar as representações dos dados espaço-temporais com as funcionalidades de consultas e processamentos dentro do próprio SGBD.

Este trabalho está estruturado da seguinte forma: o capítulo 2 aborda os conceitos do modelo de dados espaço-temporal propostos por [5], o capítulo 3 apresenta os modelos no ambiente R, . O capítulo 4 descreve o modelo PostGIS-T, e o capítulo 5 conclui o trabalho.

2 Revisão Teórica

Ferreira et. al (2014) propõe um modelo de dados espaço-temporal utilizando um formalismo algébrico. Álgebras descrevem os tipos de dados e suas operações de uma maneira formal, independentemente de linguagens de programação. A álgebra proposta é extensível, permitindo definir tipos de dados como blocos de construção para outros tipos. É preciso tomar observações como unidades básicas para a representação de dados espaço-temporal, isto permite que os usuários criem diferentes visões sobre o mesmo conjunto de observação, atendendo às necessidades de aplicação.

As observações são os nossos meios para avaliar fenômenos espaço-temporais no mundo real. Uma pesquisa recente chama a atenção para a importância de utilizar observações como base para a concepção de aplicações geoespaciais [14]. O modelo proposto define três tipos de dados espaço-temporais como abstrações construídas em *observações: séries temporais, trajetória e coverages*. A *série temporal* representa a variação de uma propriedade ao longo do tempo em um local fixo. A *trajetória* representa como localizações ou fronteiras de um objeto mudam ao longo do tempo. A *coverage* representa a variação de uma propriedade em uma extensão espacial em um determinado tempo. Também é definido um tipo de auxiliar chamado de *coverage series* que representa um conjunto ordenado em tempo de *coverages* que têm a mesma extensão. Usando esses tipos, podemos representar objetos e campos que mudam ao longo do tempo, bem como *eventos*.

2.1 Diferentes visões sobre o mesmo conjunto de observação

A Figura 1 mostra um exemplo de observações coletadas por cinco objetos. Cada observação é representada como uma tupla na forma (id, x, y, t, p) , onde id é a identificação do objeto, x e y são as localizações espaciais, t é o tempo e p é um valor de propriedade coletado na localização espacial x e y em tempo t . Neste exemplo, considera-se que a propriedade coletadas é a poluição do ar.

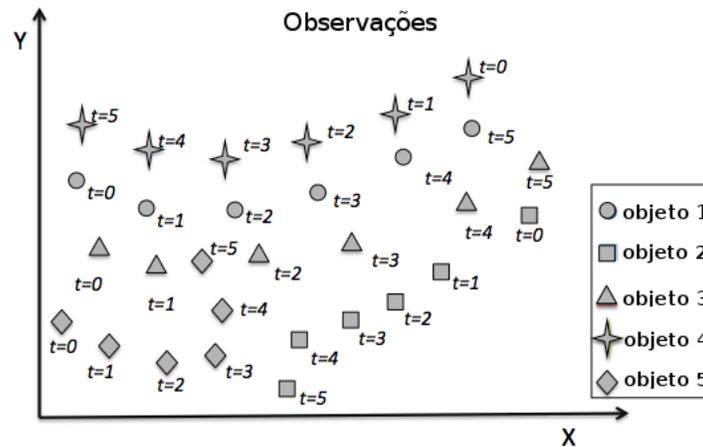


Figura 1: Observação espaço-temporal

A partir dessas observações, podem ser criadas diferentes visões para o dado espaço-temporal dependendo do tipo de análise que se deseja realizar sobre elas. Cada visão é materializada como um tipo de dados. Por exemplo, para analisar a forma como os objetos se movem ao longo do tempo e espaço, cria-se uma instância do tipo de dados *trajetória* para cada objeto. Cada instância trajetória contém observações de um objeto específico. Para analisar como a poluição do ar varia em uma região, criam-se instâncias do tipo de dados *coverage*. Cada instância do tipo coverage contém observações em um período específico, combinando observações de objetos diferentes. Para analisar como a poluição do ar varia numa dada localização espacial, pode ser criada uma instância do tipo de dados séries temporais a partir de instâncias do tipo coverage.

A Figura 2 ilustra trajetória e coverage construídas a partir do mesmo conjunto de observações apresentados na Figura 1.

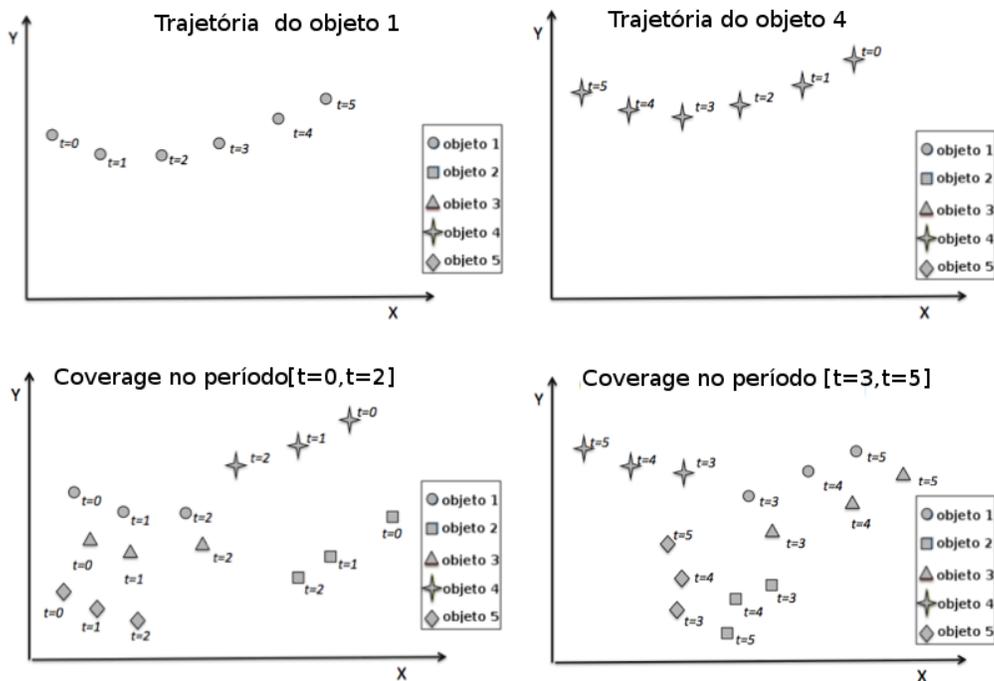


Figura 2: Diferentes tipos de dados construídos pelo mesmo conjunto de observação

Cada instância de tipo de dados espaço-temporal, *série temporal*, *trajetória* e *coverage*, possui uma função de interpolação, chamada interpolador, adequado para ele. Uma série temporal tem um interpolador que estima os valores de propriedade em tempos não medido. Uma trajetória tem um interpolador capaz de estimar localizações em tempos não-observados. A coverage tem um interpolador para estimar valores de propriedades em locais não observados.

A capacidade de criar diferentes visões ou tipos de dados com o mesmo conjunto de observação é essencial para representar adequadamente as informações espaço-temporal.

3 Manipulação e Análise de Dados Espaço-Temporal no R

Para representar os diferentes tipos de dados espaço-temporal propostos por [5] no R é necessária a integração de vários pacotes, desde pacotes de acessos até pacotes estatísticos.

Para o acesso de dados podemos utilizar pacotes como, `Rgdal`, `Rpostgres` and `Rodbc`, para os tipos de dados espaço-temporal são utilizados os pacotes `spacetime`, `xst`, `trajectories` and `raster`. O pacote `spacetime` contém como base o conjunto de tipos para representação espaço-temporal. Este conjunto é flexível no sentido em que várias representações de espaço e de tempo podem ser combinados [18]. Além disso, o R possui os pacotes estatísticos, como `gstat`, que proporcionam funções de interpolação que são cruciais para a criação de tipo de dados coverage.

Os dados espaço-temporais podem ser armazenados em diferentes fontes de dados, como banco de dados (ex. PostGis), ou arquivos de dados (ex. shapefiles e arquivos raster) e *web services* [7]. Os pacotes que acessam os dados de diferentes fontes não trabalham diretamente com o conceito de dado espaço-temporal, por exemplo, eles não conseguem identificar que o dado é do tipo trajetórias dentro do conjuntos de observações.

O pacote de acesso `Rgdal` [1], permite acessar fontes de dados como shapefiles e raster. Além disso, múltiplas camadas de dados raster podem ser acessadas através do R utilizando o pacote `Rgdal` e `raster` [11].

Para o acesso de dados a partir de um banco de dados, como por exemplo o PostGis, podemos utilizar o pacote `RODBC`. Este pacote permite selecionar subconjuntos de dados através de consultas SQL. Diferente do `Rgdal` que é um tipo de pacote que trata o dado espacial, o `RODBC` não possui este tipo de tratamento, as localizações espaciais x e y do conjunto de observações são carregadas no R como um tipo numérico ou textual, logo a conversão para o tipo espacial é necessária quando se utiliza este pacote.

Como dito anteriormente, o pacote `spacetime` é utilizado como base para representação de outros tipos de dados espaço-temporal. Para sua construção é necessário aplicar a combinação de dados espaciais e temporais em layouts, conhecidos como classes do pacote `spacetime`. Além disso, podemos atribuir valores associados a cada tempo e espaço nestes layouts, utilizando o objeto `data.frame` do R.

O `data.frame` permite visualizar os dados como uma matriz de linhas de

observações sobre colunas de valores das variáveis de interesse. A representação dos valores das variáveis de interesse podem incluir tipos numéricos de ponto flutuante ou inteiros, tipos lógicos e caracteres [2].

Para armazenar dado temporal nas classes do *spacetime*, utilizam-se os objetos da classe *xts* do pacote *xts* [22]. Este pacote foi escolhido por [18] porque ele tem um suporte para representar vários tipos de data ou hora, além disso ele é integrado com o pacote *zoo*, que possui boas ferramentas para agregação ao longo do tempo [26].

O dado espacial é a criado a partir do pacote *sp*. Este tipo de dado possui dois *slots*, um contendo o mínimo retângulo envolvente (*bounding box*) e uma matriz com as coordenadas numéricas, outro *slot* contendo uma classe que define o sistema de coordenadas de referência [2]. Os dados espaciais no R podem ser representados como pontos, linhas, polígonos ou conjunto de polígonos, ou um pixel (grade ou célula de um raster) [18], além disso as tipos de dados do pacote espacial podem ser estendida, utilizando um *data.frame* que carregando atributos associados com as geometrias. Os tipos de dados do pacote espacial são mostrados na Figura 3.

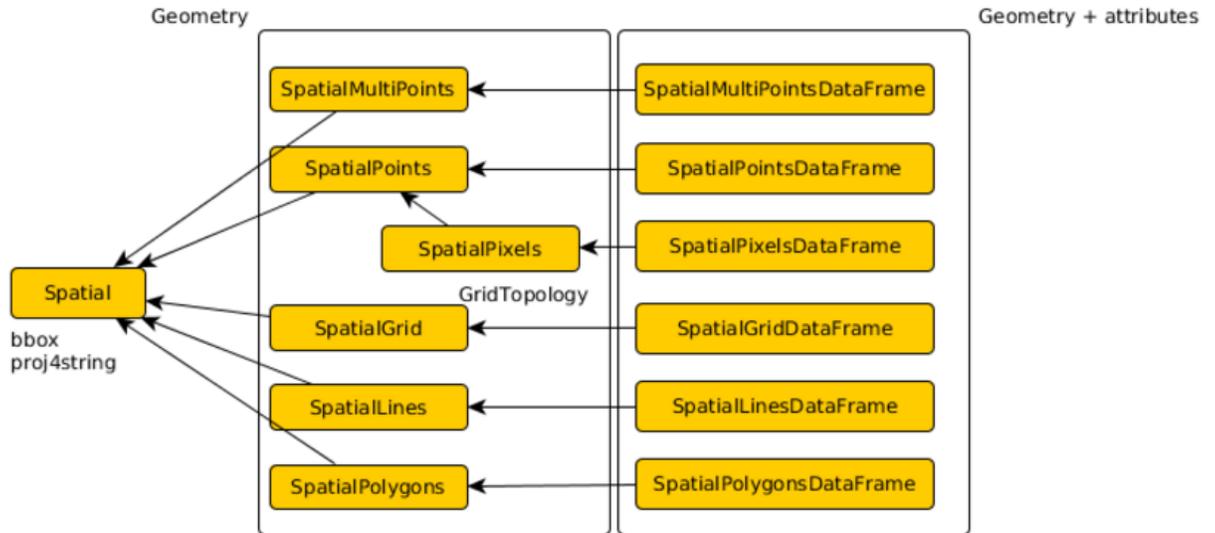


Figura 3: Tipos de dados do pacote espacial do R. [19]

A seguir serão apresentados as layouts do pacote *spacetime* para representação do dado espaço-temporal. O primeiro layout é o *Spatiotemporal*

Full Grid (STF), para cada dado espacial, uma mesma sequência temporal é amostrada [18]. Neste layout podemos representar uma série de imagens de satélites da mesma região combinado com sua resolução temporal. Outro layout, é o *Sparse grid layout* (STS), ele possui o mesmo layout do STF, porém, o STS não aceita pontos em sua grade onde não exista valores referentes aquele espaço e tempo, enquanto que o STF aceita. O último layout é o irregular (STI), neste caso para cada dado espacial existe um dado temporal [18].

Todos os layouts apresentados derivam de uma classe abstrata, espaço-temporal, chamada ST. A classe ST deriva duas ordens de classes: a classes de espaço-temporais geométricas, apresentadas anteriormente e uma classe aumentada com dados reais, sob a forma de data.frames. A Figura 4 apresenta as classes de dados espaço-temporais do pacote spacetime

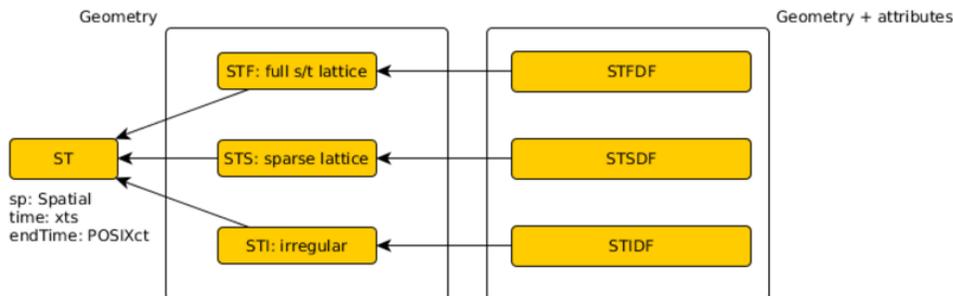


Figura 4: Classes do pacote spacetime. [19]

A partir destas classes juntamente com iteração de outros pacotes, tipos de dados espaço-temporais propostos por [5] podem ser criados.

Utilizando o pacote `trajectories` [20], podemos representar a trajetória de um objeto. Podemos extrair a série temporal de uma propriedade em uma determinada localização a partir do pacote `xts` [22]. Além disso a necessidade de utilizar um interpolador espacial ou espaço-temporal do pacote `gstat` [9] para representar uma coverage.

3.1 Estudo de caso

Nesta seção, iremos apresentar o uso dos pacotes do R apresentados na seção anterior abordando os tipos de dados propostos por [6].

O conjunto de dados utilizados para este estudo de caso contém dados de dados trajetória de navios ao longo da costa brasileira. Estas observações são armazenadas em um banco de dados PostGIS e contém trajetórias de 993 navios recolhidos durante 3 anos, de 2008 a 2011.

Para este trabalho, selecionamos um pequeno subconjunto das observações das embarcações, mostrada na Figura 5. Filtramos os dados de trajetórias temporalmente, utilizando apenas as trajetórias dos navios em um determinado dia, e espacialmente, utilizando a costa do estado do Rio de Janeiro. Os dados de observações contém um ID de embarcação (tipo inteiro), tempo (tipo timestamp), localização espacial (tipo de geometria) e velocidade de cada navio (tipo numérico) associado ao tempo e espaço. O conjunto de observações é apresentado na Figura 6.

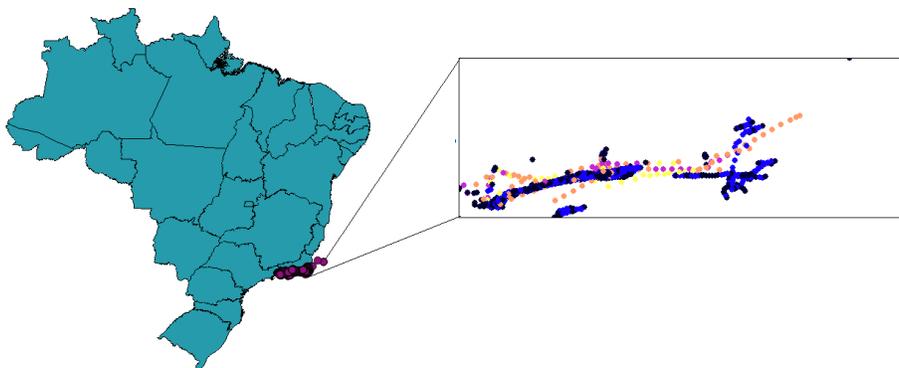


Figura 5: Visualização dos dados Filtrados por Tempo e Região

id integer	datahora timestamp without time zone	ponto geometry(Geometry,4326)	velocity numeric
583	2010-06-17 05:08:35	0101000020E6100000BBB88D06F08E45C0371AC05B201137C0	0.408763472410365
583	2010-06-17 06:08:05	0101000020E6100000A2B437F8C29445C0F90FE9B7AF1337C0	1.35374278395259
583	2010-06-17 07:08:43	0101000020E6100000287E8CB96B9145C0FBCBEEC9C31237C0	0.751117663040952
583	2010-06-17 08:08:10	0101000020E61000007958A835CD8B45C0569FABADD80F37C0	1.31993573146551
583	2010-06-17 09:09:25	0101000020E6100000BB5A679C79145C08A1F63EE5A1237C0	1.34078049066134
583	2010-06-17 10:09:45	0101000020E610000054742497FF9045C052B81E85EB1137C0	0.181341567758016

Figura 6: Conjunto de observações no PostgresSQL

Para acessar este conjunto de observação, foi utilizado pacote RODBC.

Utilizando este pacote conseguimos acessar os dados de observação através de uma consulta SQL, como mostrado na Figura 7.

```
library(RODBC)

con <- odbcConnect("PostgreSQL30")

query <- "select id, datahora, velocity, st_x(ponto) x, st_y(ponto) y
         from onedayVelocidadeAll
         order by datahora"

vesselsObs <- sqlQuery(con, query)
```

Figura 7: Acessando os dados de observação no banco de dados.

Quando as observações são selecionados pelo pacote `RODBC`, algumas mudanças devem ser feitas. Dados espaciais como latitude e longitude não são carregados no R como `SpatialPoints` do tipo espacial, neste caso, coordenadas de dados foram carregadas no R como tipo numérico, portanto, é necessário converter esses dados para tipo espacial do R, além disso é necessário adicionar o sistema de coordenadas de referência das observações utilizadas.

Para dados temporais não precisamos realizar mudanças, porque os dados foram carregados na tipo `POSIXct` do R. Este tipo é a maneira padrão de representar data e hora no R [25]. Finalmente, criamos um `data.frame` com apenas uma coluna, contendo dados de velocidade dos navios em cada instante de tempo. Combinando os objetos espaciais, temporais, e os valores do `data.frame` podemos aplicá-los na classe espaço-temporal do R. Neste caso, nós aplicamos os dados em uma grade irregular, `STIDF`, pois, nossas observações se aplicam aos tipos de dados representados por essa classe.

A partir do classe de dado espaço-temporal `STIDF`, podemos introduzir diferentes visões para este tipo de dado.

A trajetória de um navio pode ser representada, utilizando a integração do `STIDF` com classes do pacote `trajectories`. Neste caso, dois navios foram selecionados do nosso conjunto de observação, para cada navio criamos um tipo de dado espaço-temporal e em seguida utilizamos o método `Track` do pacote `trajectories` para obtermos a trajetórias destes navios. A figura 8 apresenta a trajetória realizada pelos dois navios selecionados.

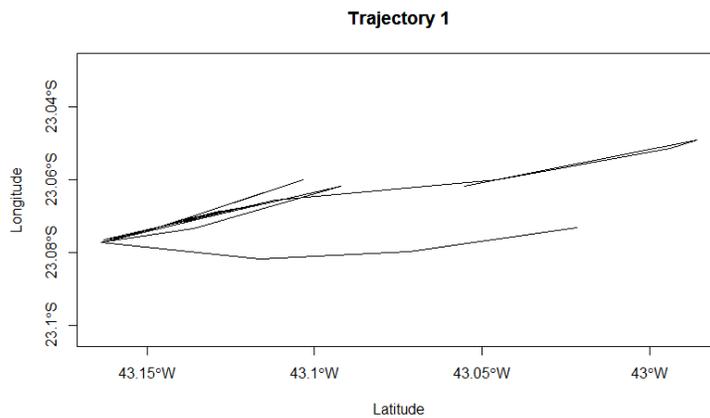


Figura 8: Trajetória do Navio 1

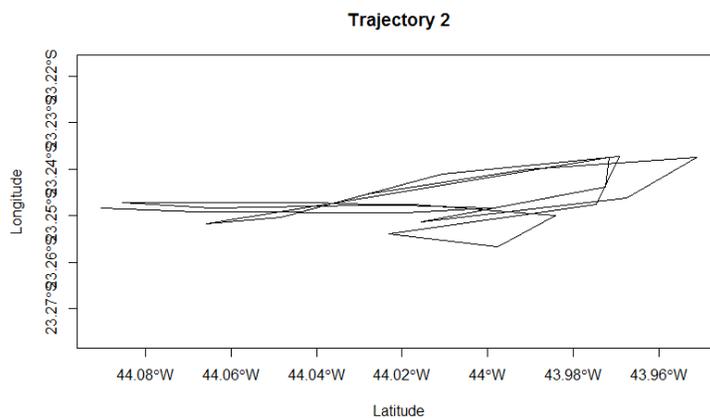


Figura 9: Trajetória do Navio 2

Além disso, podemos analisar como dois navios trafegam juntos, analisando a distância entre os dois a partir de suas trajetórias. O pacote `trajectories` permite calcular distâncias entre duas trajetórias. Uma vez comparadas duas trajetórias, nós podemos extrair uma série temporal da distância entre as duas trajetórias utilizando o pacote `xts`. A série temporal gerada pelo R da distância entre duas trajetórias é apresentada na Figura 10

Colocando juntas todas as observações de velocidades obtidas pelos navios durante um dia, nós podemos criar uma *coverage* que investiga como a

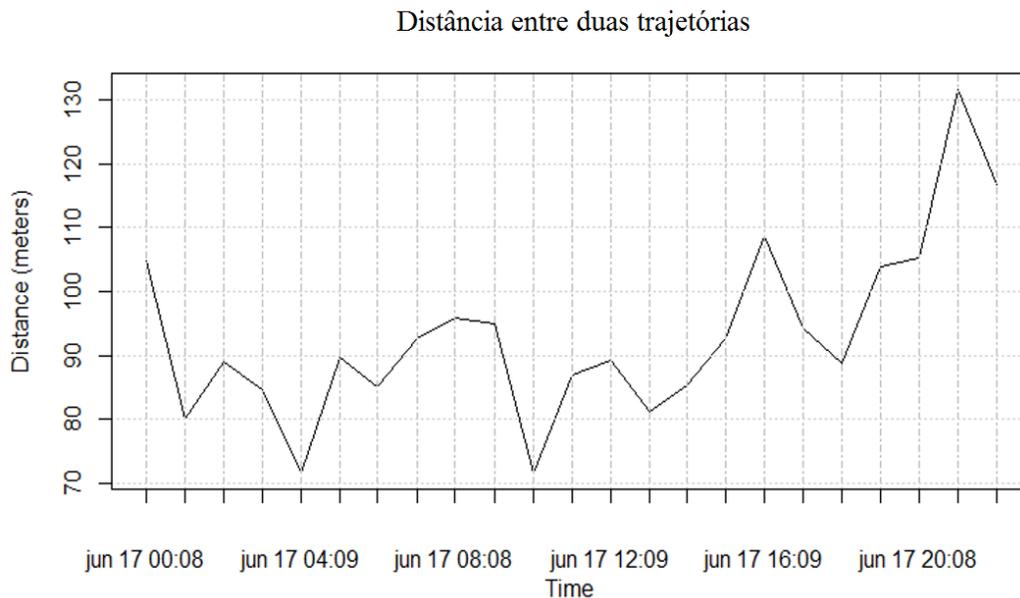


Figura 10: Distância entre duas trajetórias

velocidade varia dentro da região do nosso conjunto de observações. Este tipo de visão permite visualizar onde os navios trafegam com baixa velocidade, sendo assim, possíveis áreas de pesca.

Nossas observações são discretas, logo, elas precisam ser combinadas com funções de interpolação para aproximar a mudança contínua [5]. A interpolação espacial é utilizada neste caso, porque estamos interessados em estimar novos valores no espaço.

Por esta razão, criamos uma grade e adicionamos a ela o sistema de referência dos dados de localização dos navios, para que a interpolação seja realizada sob a mesma.

Para cada hora do dia uma interpolação foi realizada usando o interpolador Inverso da Distância ponderada. O tipo de dado retornado desta interpolação é espacial, `SpatialGridDataFrame`. Este tipo é típico para representação raster em GIS [2]. Usando `Rgdal`, cada grade pode ser salvar como no formato TIF.

Uma vez que todas as interpolações foram geradas, podemos armazenar as imagens geradas em pilhas de raster, utilizando o pacote `raster`. Duas pilhas são geradas, uma contém todos os raster gerados a cada hora durante o

período de meia noite a meio dia e outra pilha contendo os raster gerados de meio dia a 23:59 da noite. Assim, podemos associar a cada raster o tempo o qual ele foi fixado durante a interpolação, gerando uma *coverage series*. Uma vez associados os raster e o tempo de cada raster, podemos converter esse tipo de dado para o tipo STFDF do pacote spacetime.

As figuras 11 e 12 apresentam os dados espaço-temporais como raster coverages. Estas coverages mostram como a velocidade varia a cada intervalo de tempo.

Durante intervalo de meia noite a 01:00 da manhã os navios estão parados, portanto não temos nenhuma variação neste intervalo. Para cada intervalo de tempo podemos notar que existem regiões fixas que possuem a velocidade muito baixa, podendo ser possível locais de pesca. Nenhuma técnica de mineração de dados é aplicada neste caso para extração deste tipo de informação, a coverage permite avaliar visualmente onde estas são identificadas estas regiões pesqueiras.

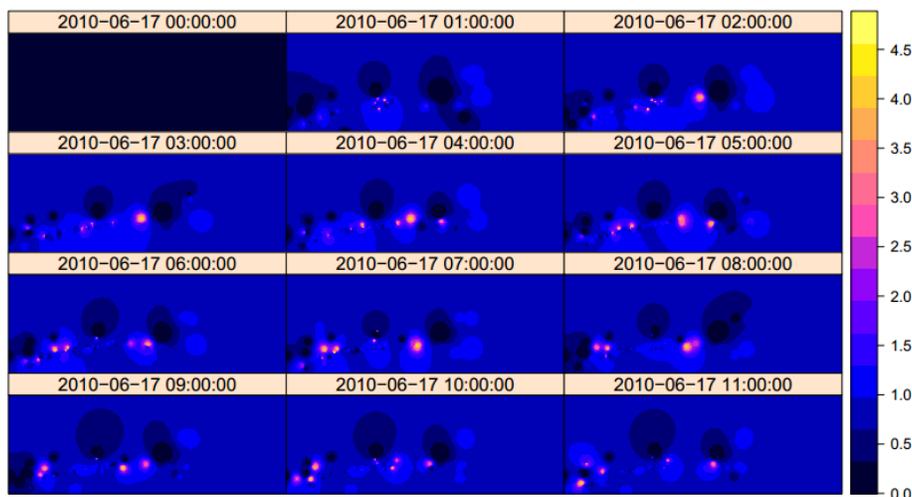


Figura 11: Variação da velocidade durante 00:00 a 12:00

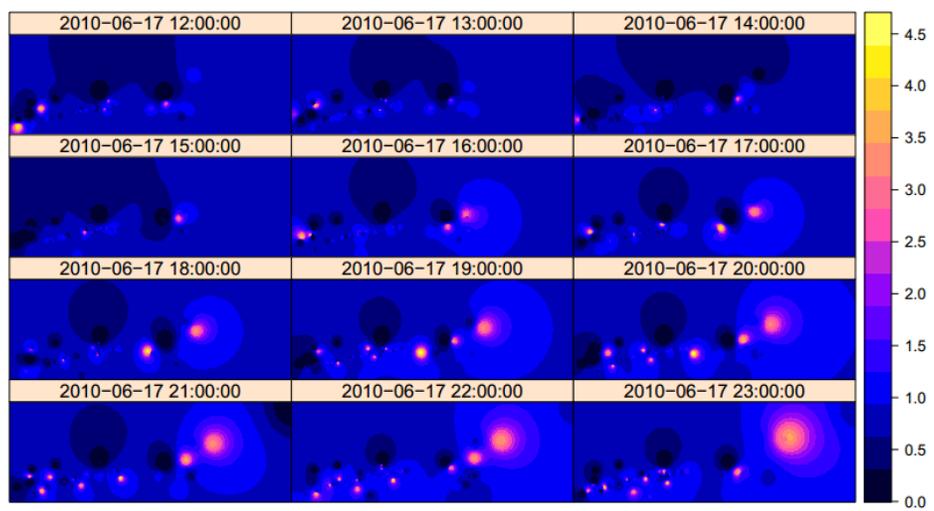


Figura 12: Variação da velocidade durante 00:00 a 23:59

4 Modelo PostGIS-T

Para representar os dados espaço-temporais no PostgreSQL, o PostGIS-T implementa o `SPATIOTEMPORAL`, um tipo composto que engloba as mesmas funções que as classes `Observations` e `SpatioTemporal`.

No PostgreSQL, o dado empírico pode ser armazenado em tuplas com uma ou mais relações. Uma vez que possuímos os dados empíricos, podemos proceder com o processo de instanciação do tipo `SPATIOTEMPORAL` por meio da função agregada `TST_SPATIOTEMPORAL()`. Essa função equivale a um construtor `new` da classe `Observation`. Note que `SpatioTemporal` é uma classe abstrata e não define um construtor, o qual deve ser fornecido pelas classes derivadas `TimeSeries`, `Trajectory` e `Coverages`. PostgreSQL opera de maneira diferente e não possui uma abordagem orientada a objeto. Nesse contexto, o tipo composto `SPATIOTEMPORAL` é definido como um container de dados gerais pelo qual é possível representar todos os tipos de dados espaço-temporais na extensão PostGIS-T.

Deve-se definir funções para que os dados possam ser acessados e manipulados. Assim, essas funções completam o nosso tipo de definição. Para atingir o mesmo grau de especificação, nós implementamos cada operador declarado no modelo definido por [5]. Além disso, para organizar nossa explicação, foram listados todas as operações de funções equivalentes na tabela 1.

type Observations: new	SPATIOTEMPORAL: TST_SPATIOTEMPORAL()
type SpatioTemporal: begins ends boundary after, before, during intersection, difference	TST_BEGINS() TST_ENDS() TST_HULL() TST_AFTER(), TST_BEFORE(), TST_DURING() TST_INTERSECTION(), TST_DIFFERENCE()
type TimeSeries: value min, max less, greater, equals	TST_MEASURE() TST_MIN(), TST_MAX() TST_LESS(), TST_GREATER(), TST_BETWEEN()
type Trajectory: value	TST_LOCATION()
type Coverage: value min, max less, greater, equals	TST_MEASURE() TST_MIN(), TST_MAX() TST_LESS(), TST_GREATER(), TST_BETWEEN()

Tabela 1: Lista das equivalências entre todas as operações e funções.

Além das funções acima comparadas, outras foram desenvolvidas: `TST_OBSERVATIONS()`, `TST_RESAMPLE_TIME()` e `TST_COVERAGE()`. A primeira retorna a quantidade de observações de um tipo `SPATIOTEMPORAL`, a segunda realiza uma reamostragem temporal de séries temporais e de trajetórias. O terceiro calcula um espaço celular para obtenção de uma coverage.

4.1 Global Drifter Program: um exemplo de aplicação

Levando em consideração os dados espaço-temporais e a sua complexidade, nós escolhemos os dados de boias (*drifter*) para exemplificar as possibilidades de implementação de uma extensão para DBMS Postgres/PostGIS. Ressaltamos

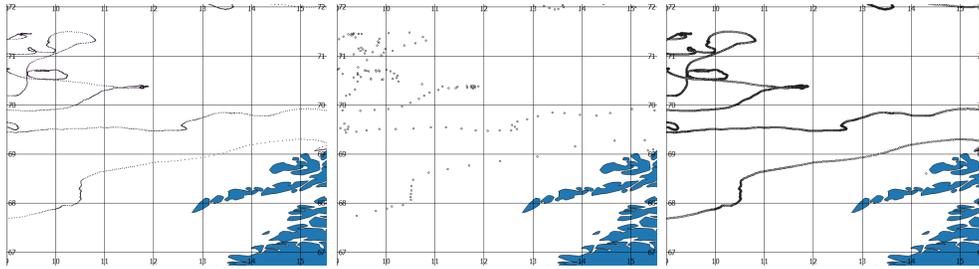


Figura 13: Reamostragem da trajetória. Da esquerda para a direita: dado original, reamostragem com dados de menor resolução temporal, reamostragem com dados de maior resolução temporal.

que ao longo desse trabalho, usaremos o termo *drifter* como sinônimo de boias, já que se referem ao mesmo objeto. As boias fazem parte do Programa Internacional Drifter (GDP) e é uma das vertentes do NOAA (*National Oceanic & Atmospheric Administration*), tendo por objetivo manter um monitoramento global contínuo das boias. Rastreadas por satélite, as boias utilizadas no sistema possuem alguns sensores, bem como bateria para o envio de seus dados. Assim, os drifters produzem uma grande quantidade de dados sobre correntes, temperatura da superfície de oceanos, pressão atmosférica, ventos e salinidade, com resolução de um grau [16]. Esses dados são, dessa maneira, de grande relevância para o desenvolvimento de pesquisas relacionadas a temas como movimento de correntes oceânicas e o clima.

Para a aplicação desse trabalho, a base de dados dos drifters usados contém 2, 263, 842 registros, com resolução de 1 hora, podendo ser acessados em [4].

Os códigos SQL a seguir mostram como podemos obter diferentes reamostragens temporais das observações. Os resultados correspondentes podem ser verificados na Figura 13.

```
-- Original data
SELECT buoy_id, spatiotemp
  FROM buoy_obs_st
 WHERE buoy_id = 11915950

-- Sub-sampling
SELECT buoy_id, TST_RESAMPLE_TIME(spatiotemp,
TST_OBSERVATIONS(spatiotemp) / 10, 'LINEAR')
  FROM buoy_obs_st
```

```

WHERE buoy_id = 11915950

-- Oversampling
SELECT buoy_id, TST_RESAMPLE_TIME(spatiotemp,
TST_OBSERVATIONS(spatiotemp) * 2, 'LINEAR')
FROM buoy_obs_st
WHERE buoy_id = 11915950

```

O primeiro exemplo refere-se ao dado original da trajetória de uma boia enquanto que o segundo e o terceiro exemplos representam uma reamostragem dos dados originais utilizando interpolador linear. Isso denota potenciais aplicações dessa extensão para obtenção de dados interpolados.

O código seguinte ilustra como podemos obter uma coverage. Os parâmetros '7' e '13' da função TST_COVERAGE() denotam as dimensões da grade criada da coverage, que pode ser visualizada na Figura 14. Os dados usados para a medida foi a velocidade média obtida em cada uma das células.

```

-- Coverage
SELECT buoy_id, TST_COVERAGE(spatiotemp,
TST_HULL(spatiotemp), 7, 13, 'AVG')
FROM buoy_obs_st
WHERE buoy_id = 132470

```

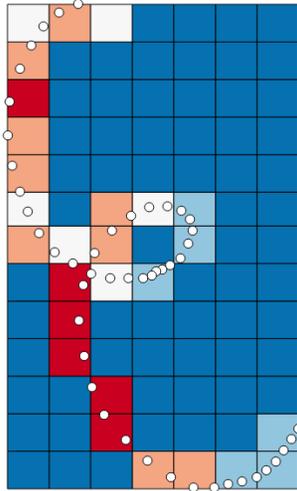


Figura 14: Coverage calculada a partir da velocidade média das boias. As cores mais quentes representam maior velocidade das boias.

5 Conclusão

Embora existam diversos esforços que visem o tratamento de dados espaço-temporais, ainda não foi possível desenvolver metodologias consolidadas ou padrões oficiais de representação. Foram aqui destacados duas abordagens de tratamento de dados espaço-temporais, em linguagem de programação R e em PostgreSQL. Assim, embora preliminares, algumas observações devem ser realizadas. Atualmente, a linguagem R é comumente utilizada no meio acadêmico, denotando assim uma potencialidade para o seu desenvolvimento e constante aperfeiçoamento. Entretanto, o mesmo não ocorre para a abordagem em PostgreSQL, o qual é voltado principalmente para os usuários da extensão PostGIS. Apesar disso, a principal vantagem desta última abordagem se refere ao processamento de dados de grande volume, variedade e velocidade, também conhecidos por *big data*, sem a necessidade de transferência massiva de dados para o seu processamento. As duas implementações foram baseadas no modelo conceitual proposto por Ferreira et al. (2014). Todavia, diferenças puderam ser identificadas. Na primeira aplicação, uma desvantagem é não ter um pacote que acessa os dados diretamente no tipo espaço-temporal. Assim, é necessário preparar os dados espaciais e temporais separadamente para depois aplicá-los na classe espaço-temporal. Outra desvantagem é a

respeito dos pacotes que fazem acesso ao banco de dados, eles por exemplo não tratam os dados do tipo espacial, então primeiramente, uma conversão para este tipo espacial deve ser realizada. No entanto, existem pontos positivos de trabalhar com dados espaço-temporal no R. A variedade de pacotes que trabalham com dados espaço-temporal permitem representar um tipo de dado espaço-temporal em diferentes visões. Além disso existem pacotes que trazem uma grande variedade de interpoladores espaciais, que são fundamentais para criação de uma coverage, bem como pacotes que suportam dados estatísticos, permitindo análises de dados espaço-temporais.

Na aplicação voltada ao PostgreSQL, como foi desenvolvido um protótipo, ainda há a necessidade de desenvolvê-lo em outra linguagem, como *C*. Além disso, há uma necessidade em analisar quais são as melhores formas de armazenamento para que o emprego do Postgres seja consolidado, considerando as suas diversas aplicações potenciais. Uma das possibilidades para o aperfeiçoamento do sistema seria inverter a forma de apresentação dos dados, de tuplas, para colunas. As tuplas são viáveis quando objetiva-se analisar as bóias independentemente. Entretanto, atributos referentes a um conjunto de bóias poderiam ser analisados de maneira mais eficiente, caso os dados fossem armazenados em colunas. Além do mais, com esse tipo de aplicação sobre a integração de dados espaço-temporais de bóias de monitoramento, poderiam servir de base para inúmeras pesquisas, como observações das correntes oceânicas e massas de água, temperatura da superfície do mar, a pressão atmosférica, ventos e salinidade ao longo do tempo e espaço.

Por fim, destacamos que embora esses avanços tenham sido realizados, ainda não há uma solução definitiva para a complexidade envolvida no processamento de dados espaço-temporais. Nessa vertente, estudos mais pormenorizados são requeridos para que modelos possam ser incorporados, garantindo assim a interoperabilidade entre sistemas e extração de dados espaço-temporais.

Referências

- [1] Roger Bivand, Tim Keitt, and Barry Rowlingson. Rgdal: Bindings for the geospatial data abstraction library. r package version 0.8-10. 2013.
- [2] Roger S. Bivand, Edzer Pebesma, and Virgilio Gomez-Rubio. *Applied spatial data analysis with R, Second edition*. Springer, NY, 2013.
- [3] Gilberto Camara, Max J Egenhofer, Karine Ferreira, Pedro Andrade, Gilberto Queiroz, Alber Sanchez, Jim Jones, and Lubia Vinhas. Fields as a generic data type for big spatial data. In *International Conference on Geographic Information Science*, pages 159–172. Springer, 2014.
- [4] Shane Elipot, Rick Lumpkin, Renellys C Perez, Jonathan M Lilly, Jeffrey J Early, and Adam M Sykulski. A global surface drifter data set at hourly resolution. *Journal of Geophysical Research: Oceans*, 2016.
- [5] Karine Reis Ferreira, Gilberto Camara, and Antônio Miguel Vieira Monteiro. An algebra for spatiotemporal data: From observations to events. *Transactions in GIS*, 18(2):253–269, 2014.
- [6] Karine Reis Ferreira, Gilberto Camara, and Antonio Miguel Vieira Monteiro. An algebra for spatiotemporal data from observations to events. *Transactions in GIS*, 18(2):253–269, 2014.
- [7] Karine Reis Ferreira, André Gomes de Oliveira, Antônio Miguel Vieira Monteiro, and Diego Benincasa de Almeida. Temporal GIS and spatiotemporal data sources. In *XVI Brazilian Symposium on GeoInformatics(GEOINFO), Campos do Jordão, São Paulo, Brazil, November 29 - December 2, 2015.*, pages 1–13, 2015.
- [8] Antony Galton and Riichiro Mizoguchi. The water falls but the waterfall does not fall: New perspectives on objects, processes and events. *Applied Ontology*, 4(2):71–107, 2009.
- [9] Benedikt Graler, Edzer Pebesma, and Gerard Heuvelink. Spatio-temporal interpolation using gstat. 2016.
- [10] Ralf Hartmut Gutting and Markus Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.

- [11] Robert J. Hijmans. Introduction to the raster package. 2016.
- [12] K. Hornsby and M. Egenhofer. Identity-based change: a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14(3):207–224, 2000.
- [13] ISO. Geographic information - schema for moving features. ISO 19141:2008, International Organization for Standardization, Geneva, Switzerland, 2008.
- [14] Werner Kuhn. *A Functional Ontology of Observation and Measurement*, pages 26–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [15] Y. Liu, Michael F. Goodchild, Q. Guo, Y. Tian, and L. Wu. Towards a general field model and its order in GIS. *International Journal of Geographical Information Science*, 22(6):623–643, 2008.
- [16] Rick Lumpkin and Mayra Pazos. Measuring surface currents with surface velocity program drifters: the instrument, its data, and some recent results. In A. J. Mariano T. Ozgokmen A. Griffa, A. D. Kirwan and T. Rossby, editors, *Lagrangian analysis and prediction of coastal and ocean dynamics*, chapter 2, pages 39–67. Cambridge University Press New York, NY, 2007.
- [17] OGC. Opengis implementation specification for geographic information - simple feature access-part 1:common architecture. Technical Report 19141:2008, OPEN GEOSPATIAL CONSORTIUM, Geneva, Switzerland, 2006.
- [18] Edzer Pebesma. spacetime: Spatio-temporal data in R. *Journal of Statistical Software*, 51(7):1–30, 2012.
- [19] Edzer Pebesma. Handling and analyzing spatial, spatiotemporal and movement data, jun 2016.
- [20] Edzer Pebesma and Benedikt Klus. Analysing trajectory data in R. 2015.
- [21] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

- [22] JA Ryan and JM Ulrich. xts: extensible time series. r package version 0.8-6. 2012.
- [23] Michael F. Worboys. A unified model for spatial and temporal information. *Comput. J.*, 37(1):36–34, 1994.
- [24] Michael F. Worboys. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19(1):1–28, 2005.
- [25] Diethelm Wuertz, Tobias Setz, Yohan Chalabi, and Martin Maechler and Joe W. Byers. Package timedate. 2015.
- [26] Achim Zeileis and Gabor Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005.