Objetos Móveis e Trajetórias

Karine Reis Ferreira

Workshop em Processamento de Imagens para Aplicações em VANTs

21 de junho de 2012

Objetivo

(1) Estudo/Pesquisa sobre representação, consultas/operações e visualização de **objetos móveis** e **trajetórias**.

(2) Implementação de um módulo de software na TerraLib5 para tratar **objetos móveis** e **trajetórias**.

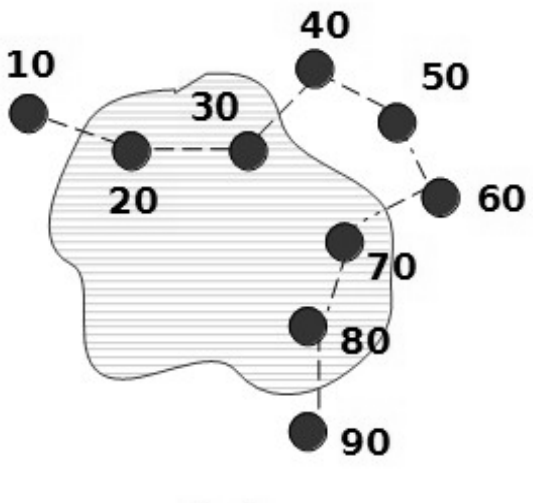
Moving Objects - Conceito

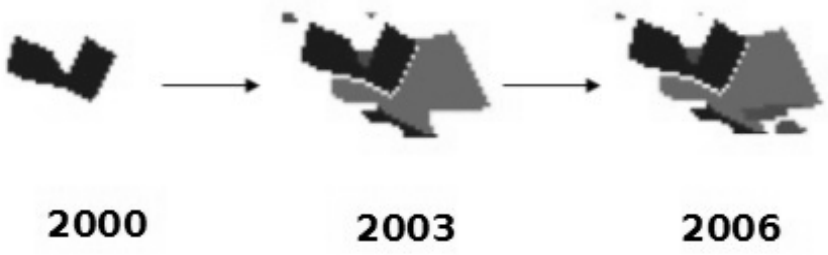
Moving Objects é um conceito já bem estabelecido e conhecido em “GIS science”.

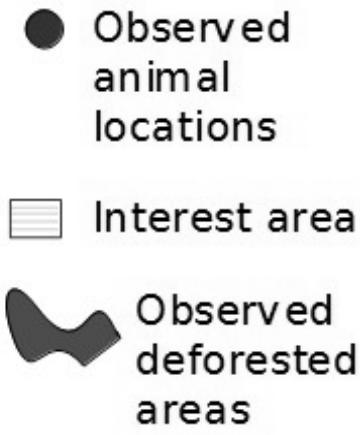
Moving Objects are entities whose spatial positions or extents change continuously over time (Guting and Schneider, 2005).

Moving Objects - Exemplos

Moving Objects are entities whose spatial positions or extents change continuously over time (Guting and Schneider, 2005).







Animal tracking

↓

moving point

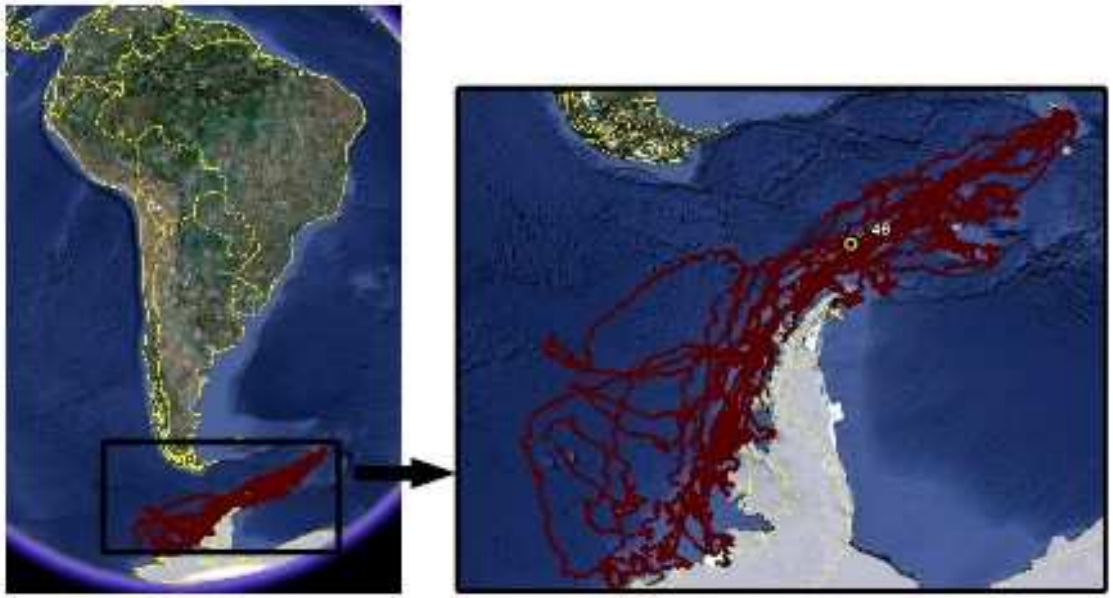
Deforested region evolution

↓

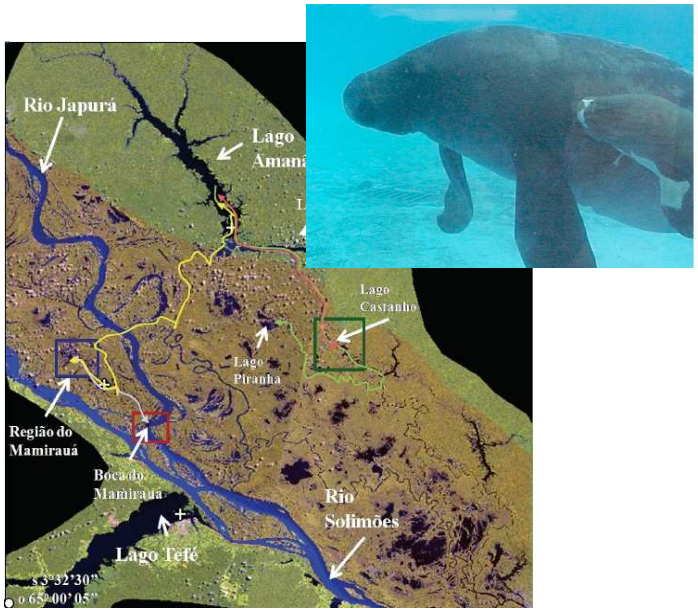
moving region

Moving Objects - Exemplos

Moving Objects are entities whose spatial positions or extents change continuously over time (Guting and Schneider, 2005).



A project that monitors sea elephants in the Antarctica.



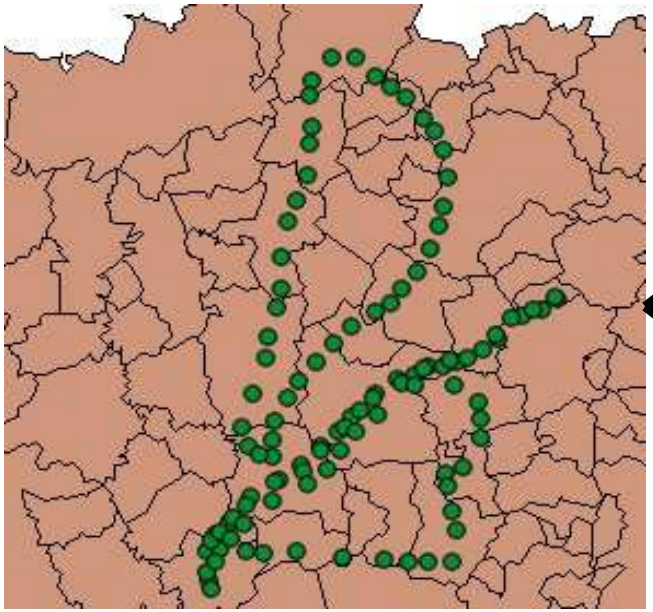
(Arraut, E. M. 2008)

Trajectory - Conceito

Trajectories are countable journeys associated to objects which are moving in space over time. (Spaccapietra et. al, 2008).

Trajectory - Exemplos

Trajectories are countable journeys associated to objects which are moving in space over time. (Spaccapietra et. al, 2008).



Quais as trajetórias diárias do carro X_1?

Quais as trajetórias do carro X_1 na zona sul da cidade?

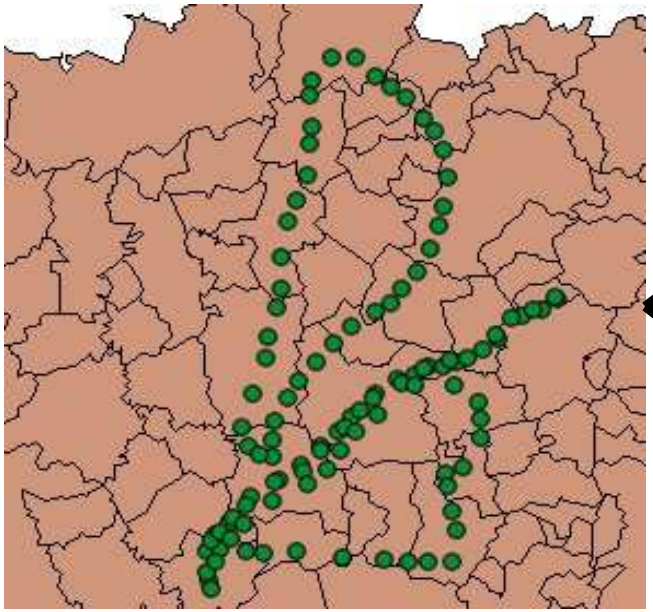
Quais as trajetórias do carro X_1 com velocidade maior que 60 km/h?

Monitoramento de carros em uma ciadade. Cada carro é um *moving object*.

Diferentes trajetórias de um mesmo *moving object*.

Trajectory - Exemplos

Trajectories are countable journeys associated to objects which are moving in space over time. (Spaccapietra et. al, 2008).



Quais as trajetórias **diárias** do carro X_1?

Quais as trajetórias do carro X_1 na **zona sul da cidade**?

Quais as trajetórias do carro X_1 com **velocidade maior que 60 km/h**?

Monitoramento de carros em uma ciadade. Cada carro é um *moving object*.

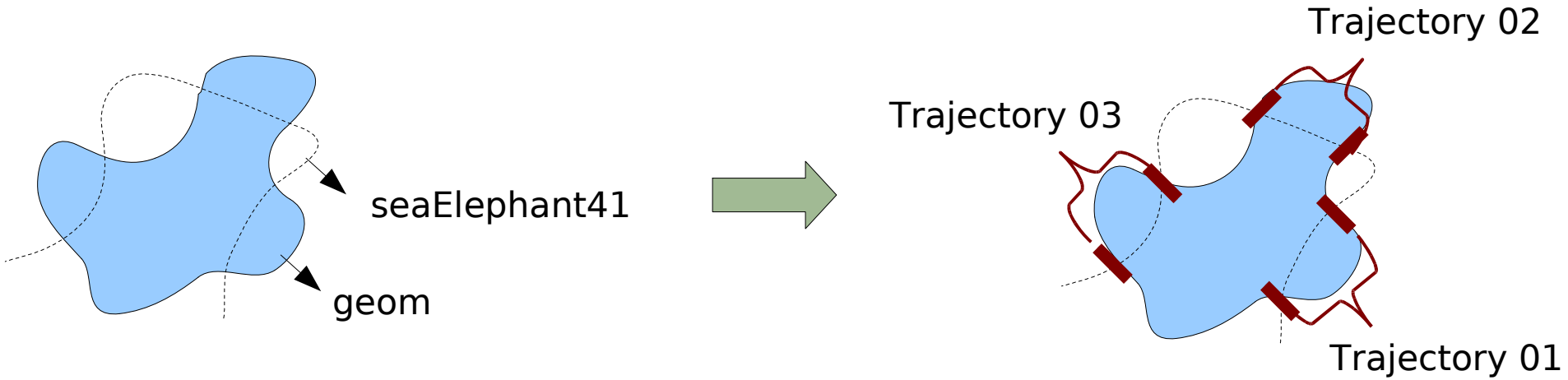
Diferentes trajetórias de um mesmo *moving object*.

Moving Objects – Algumas Operações

intersection: MovingObject x Geometry → {Trajectory}

difference: MovingObject x Geometry → {Trajectory}

intersection(seaElephant41, geom)



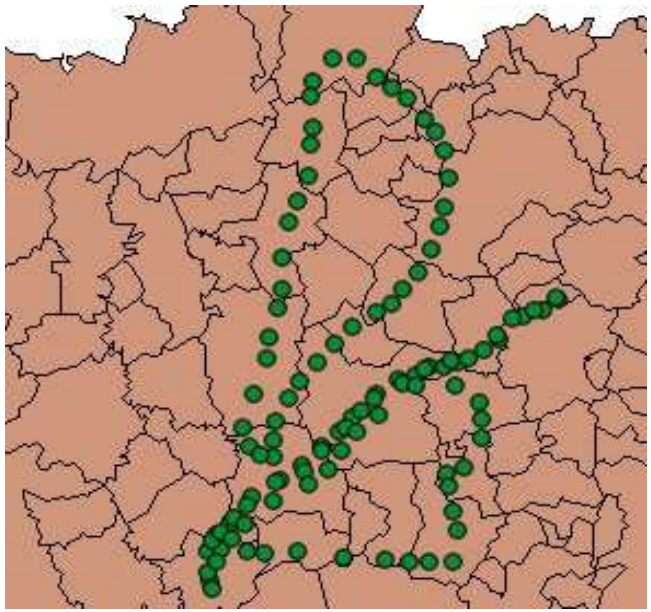
Moving Objects – Algumas Operações

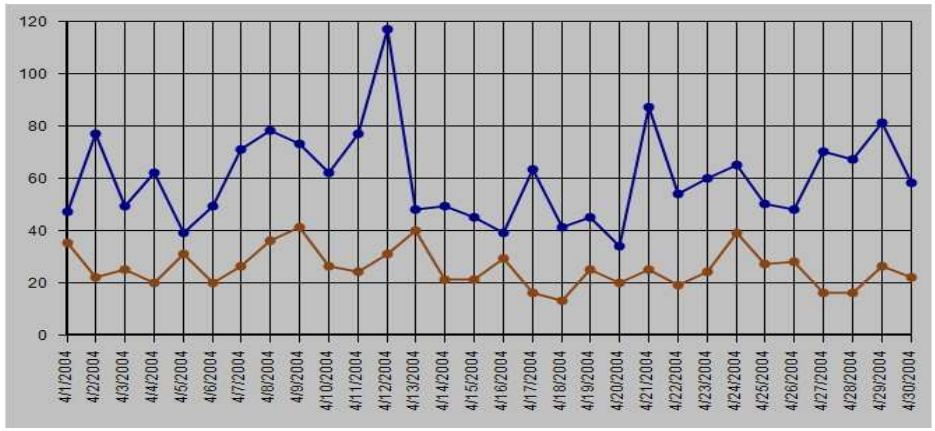
distance: MovingObject x MovingObject → TimeSeries

enters, **exits**, **reaches**, **leaves**:

MovingObject x Geometry → {Trajectory}

distance(car1, car2)





Moving Objects – Algumas Operações

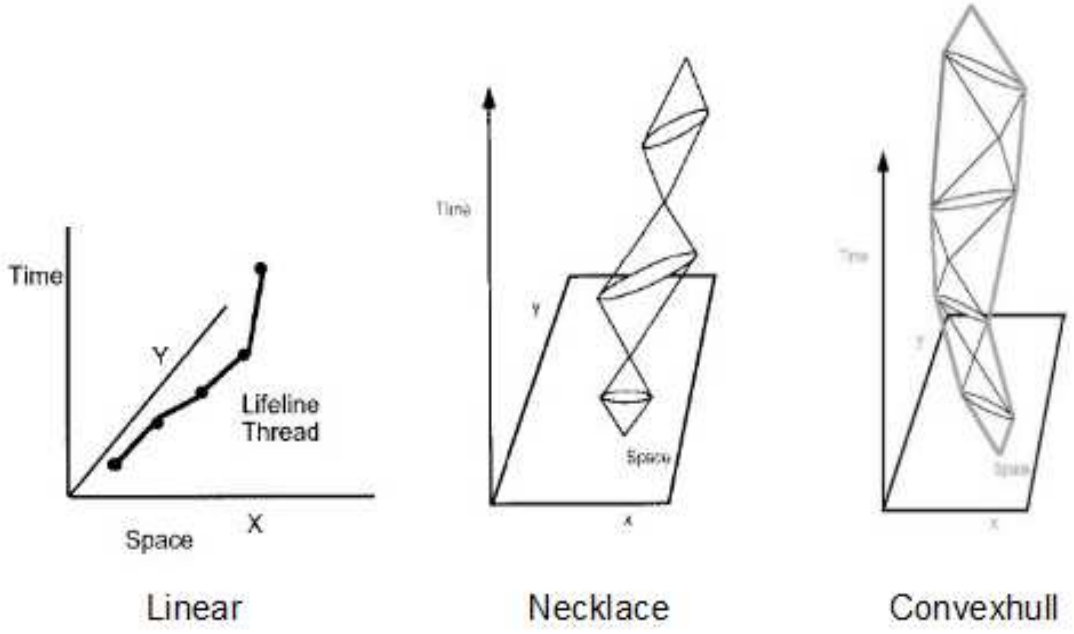
speed: MovingPoint → TimeSeries

direction: MovingPoint → TimeSeries

linearPath: MovingPoint → Line

necklacePath: MovingPoint → MultiPolygon

convexhullPath: MovingPoint → Polygon

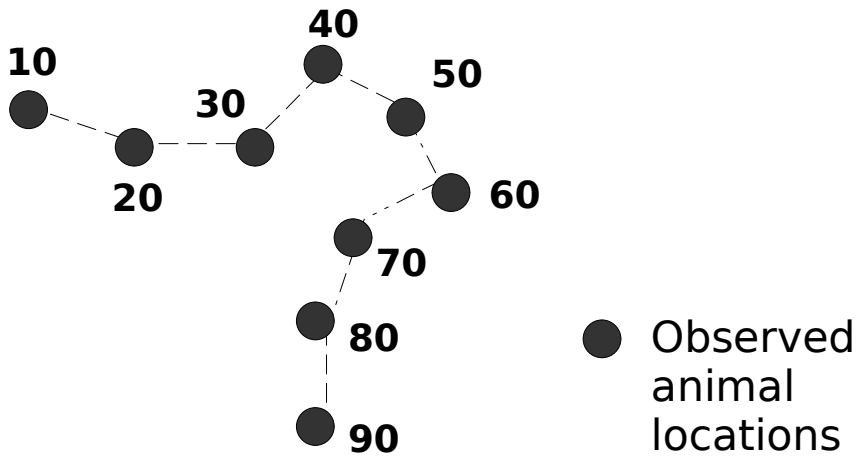


(Hornsby and Egenhofer, 2002)

Interpolation Function

Moving Object: Set of Observations + Interpolation Function

An *interpolation function* (or *interpolators*) for moving objects is a procedure that is able to estimate a spatial position or extent at any non-observed time.

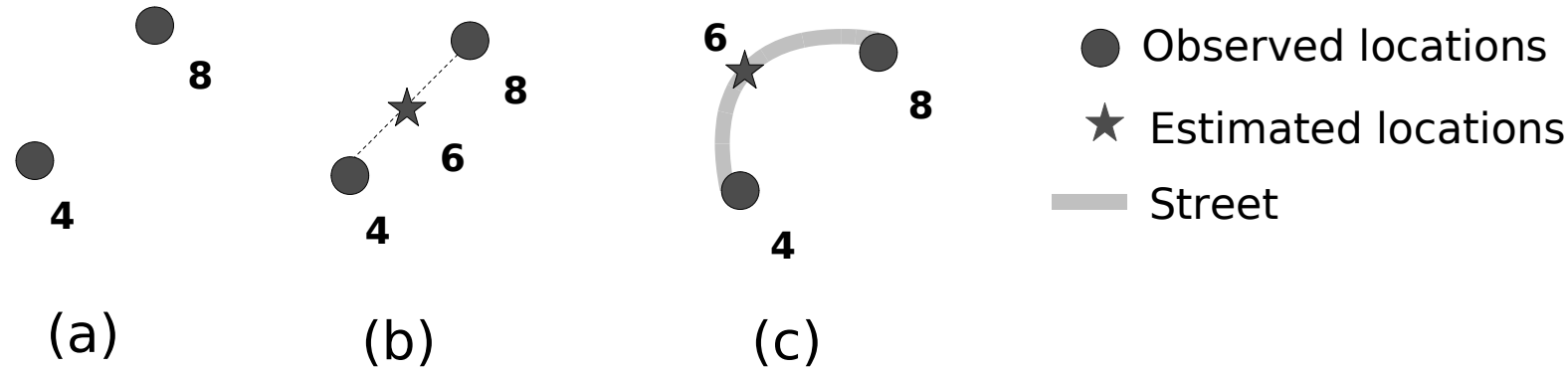


Moving Object

Where was the animal at time 55?

Interpolation Function - Examples

Different kinds of *interpolators* can be defined and used over the same set of observations.



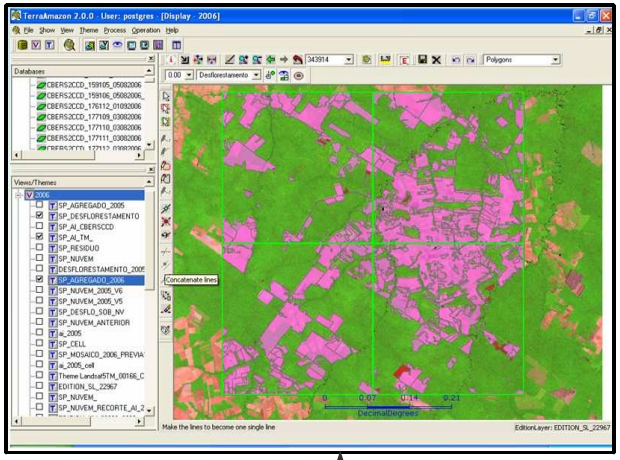
(a) given two car locations, one observed at time instant 4 and another at 8

(b) linear interpolator to estimate the non-observed time 6

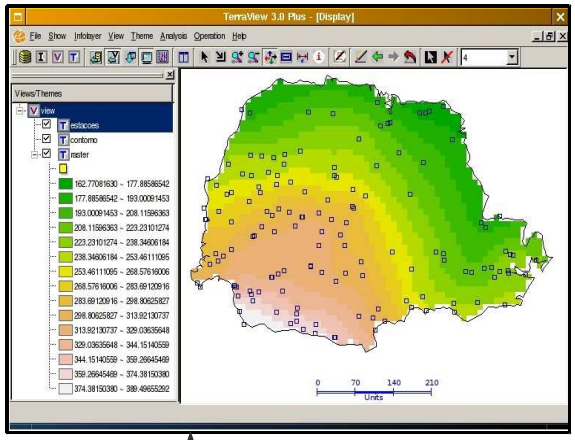
(c) interpolator which regards a street map in its estimation

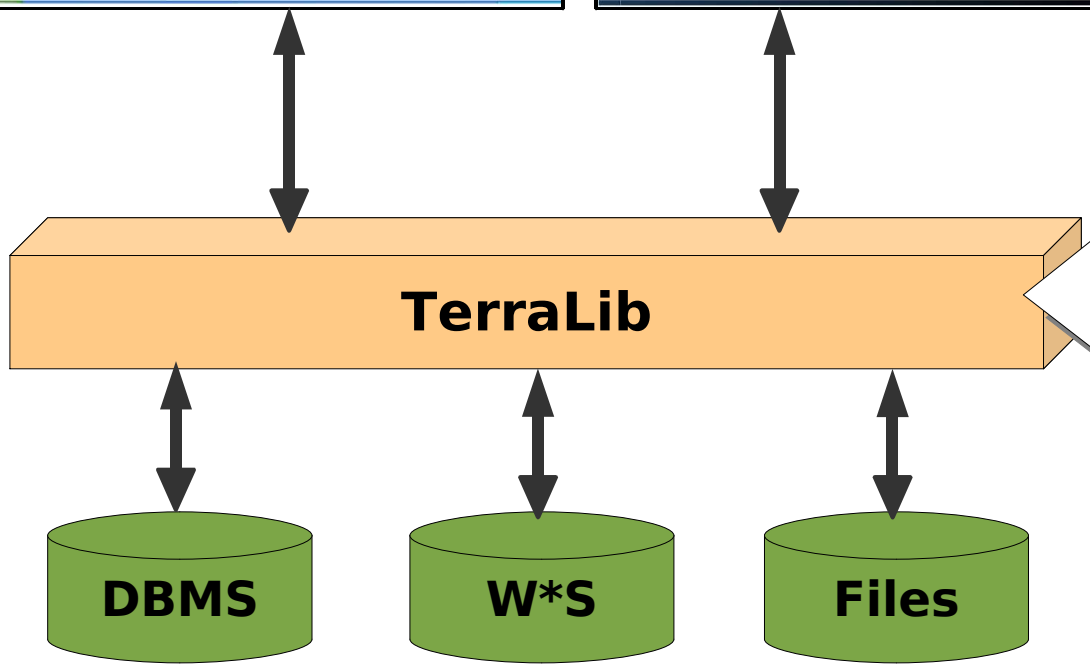
TerraLib: a FOSS geographic library

TerraAmazon



TerraView





Software library base to develop geographic information systems.

Free and Open Source Software (FOSS).

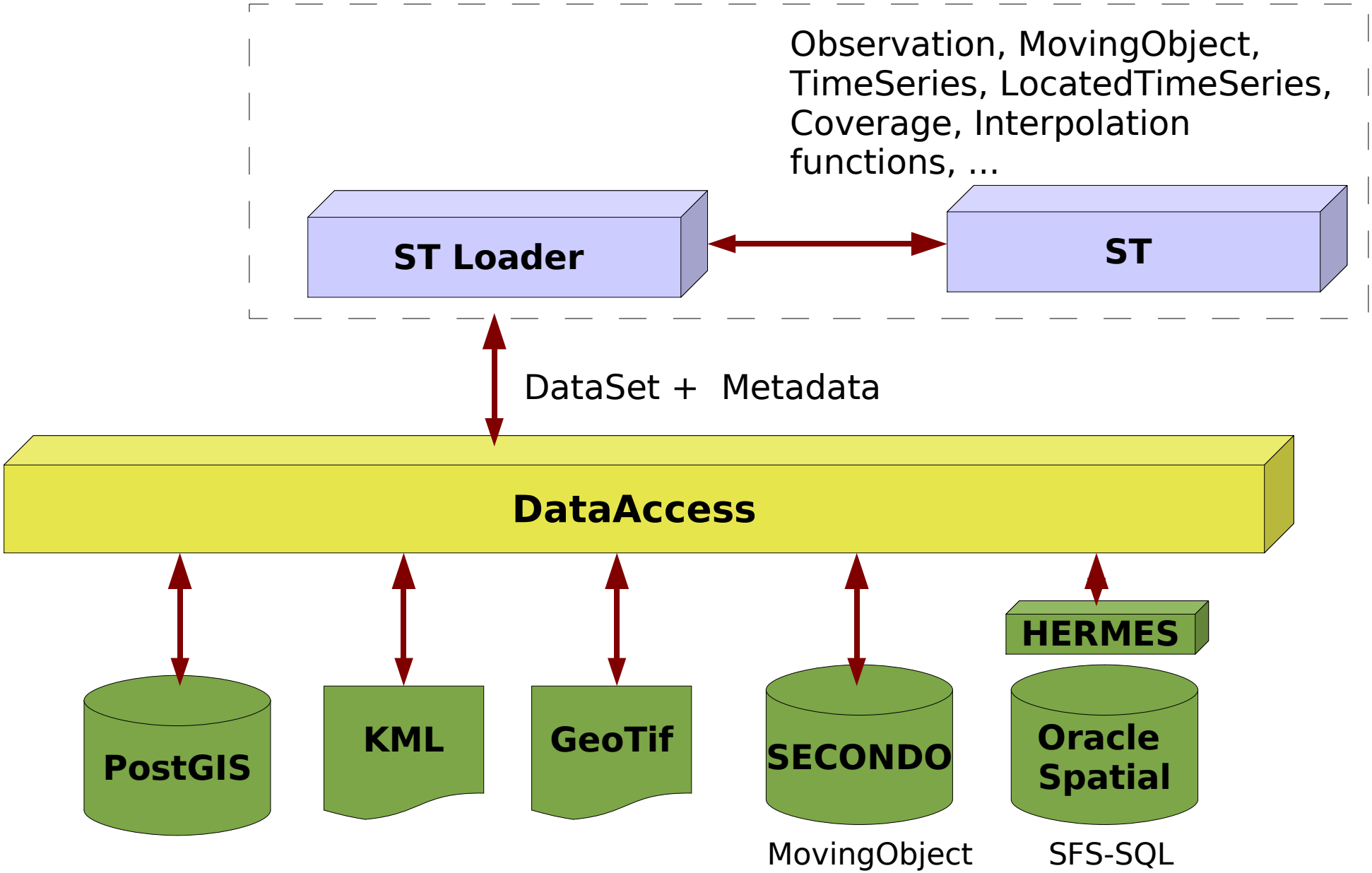
Developed by INPE.

C++ language.

Provides: spatial operations,image processing, spatial analysis, R interface, ...

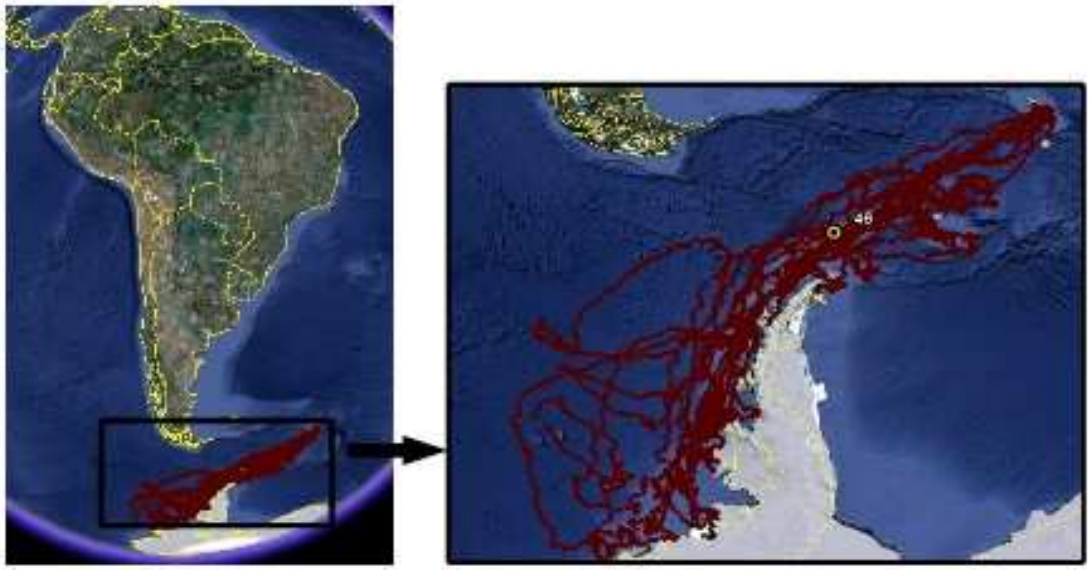
www.terralib.org

TerraLib: Modules for Spatiotemporal data



TerraLib ST Loader Module – An Example

KML file



A project that monitors sea elephants in the Antarctica.

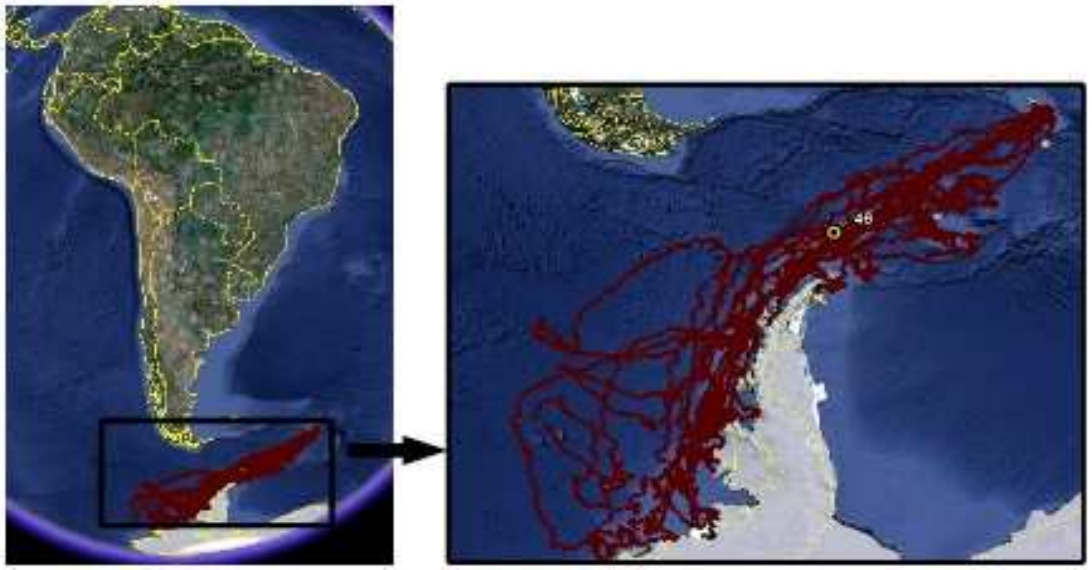
- All observations of each sea elephant: `kml::FolderType`.

- Each animal observation is represented by a `kml::PlacemarkType` type: (a) spatial location: `kml::PointType`; (b) time instant: `kml::TimeStampType`

TerraLib ST Loader Module – An Example

How to extract moving objects from KML files?

KML file



A project that monitors sea elephants in the Antarctica.

- All observations of each sea elephant: `kml::FolderType`.

- Each animal observation is represented by a `kml::PlacemarkType` type: (a) spatial location: `kml::PointType`; (b) time instant: `kml::TimeStampType`

TerraLib ST Loader Module – An Example

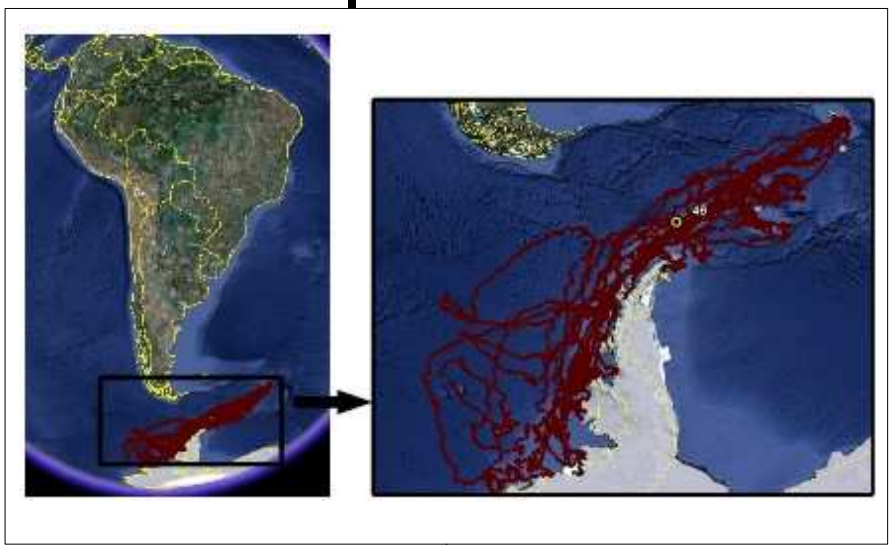
- Although KML files can be used to describe **journeys**, there is not a standard way to represent them as trajectories of moving objects for further analysis;
- Each software or mobile device that generates KML files with journeys uses its own structure for representing them;
- We can visualize journeys described in KML files in many software tools, such as Google Earth, but few of them are able to process or analyze these journeys as moving object trajectories:
  - *“When did object o1 enter a specific region r10 and how long did it stay in this region?”*

TerraLib ST Loader Module – An Example

XML metadata file

ST Loader

KML file



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<MovingObjectMetadata
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.terralib.org/movingobjectmetadata"
    xsd:schemaLocation=
    "http://www.terralib.org/movingobjectmetadata
                                movingobjectmetadata.xsd">
<kmlFileName>sea_elephants</kmlFileName>
<MovingObject>
    <name>2340</name>
    <type>MovingPoint</type>
    <ObservationContainer>
        <type>kml::FolderType</type>
        <name>40: locations</name>
    </ObservationContainer>
</MovingObject>
<MovingObject>
    <name>2341</name>
    <type>MovingPoint</type>
    <ObservationContainer>
        <type>kml::FolderType</type>
        <name>41: locations<name>
    </ObservationContainer>
</MovingObject>
</MovingObjectMetadata>
```

TerraLib: Code example

```
DataSource* ds = DataSourceFactory::make("OGR");

xmlMetadataFile = ".\\data\\kml\\sea_eleph_metadata.xml";

vector<MovingObject*> output;

DataLoader::loadMovingObjects(ds, xmlMetadataFile, output);
```

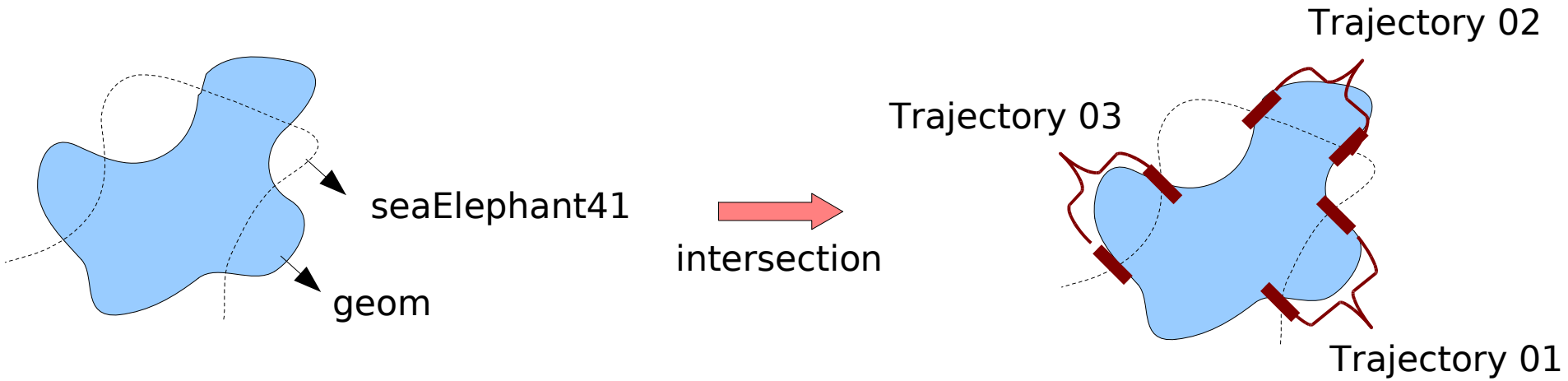
(1) OGR LIBKML Driver to read KML files
(2) Xerces-C++ to read and write XML files.

TerraLib ST module: Code example

```
MovingObject* seaElephant40 = output[0];
MovingObject* seaElephant41 = output[1];

TimeSeries* dist = seaElephant40->distance(seaElephant41);

vector<Trajectory*> trajs;
seaElephant41->intersection(geom, trajs);
```



Final remarks

- Visualization
- Patterns of trajectories
- Future

Obrigada!

Karine Reis Ferreira
(karine@dpi.inpe.br)