Smoothing methods for ARGOS trajectories

Projet long presentation

2015

Projet Long

- Project in collaboration with **CLS**
- Supervisor: **Beatriz Calmettes**, CLS
- Project leader: Jérôme Combanière
- Project team:
  - Jérôme Combanière
  - Anthony Delannoy
  - Benoit Madiot

Context

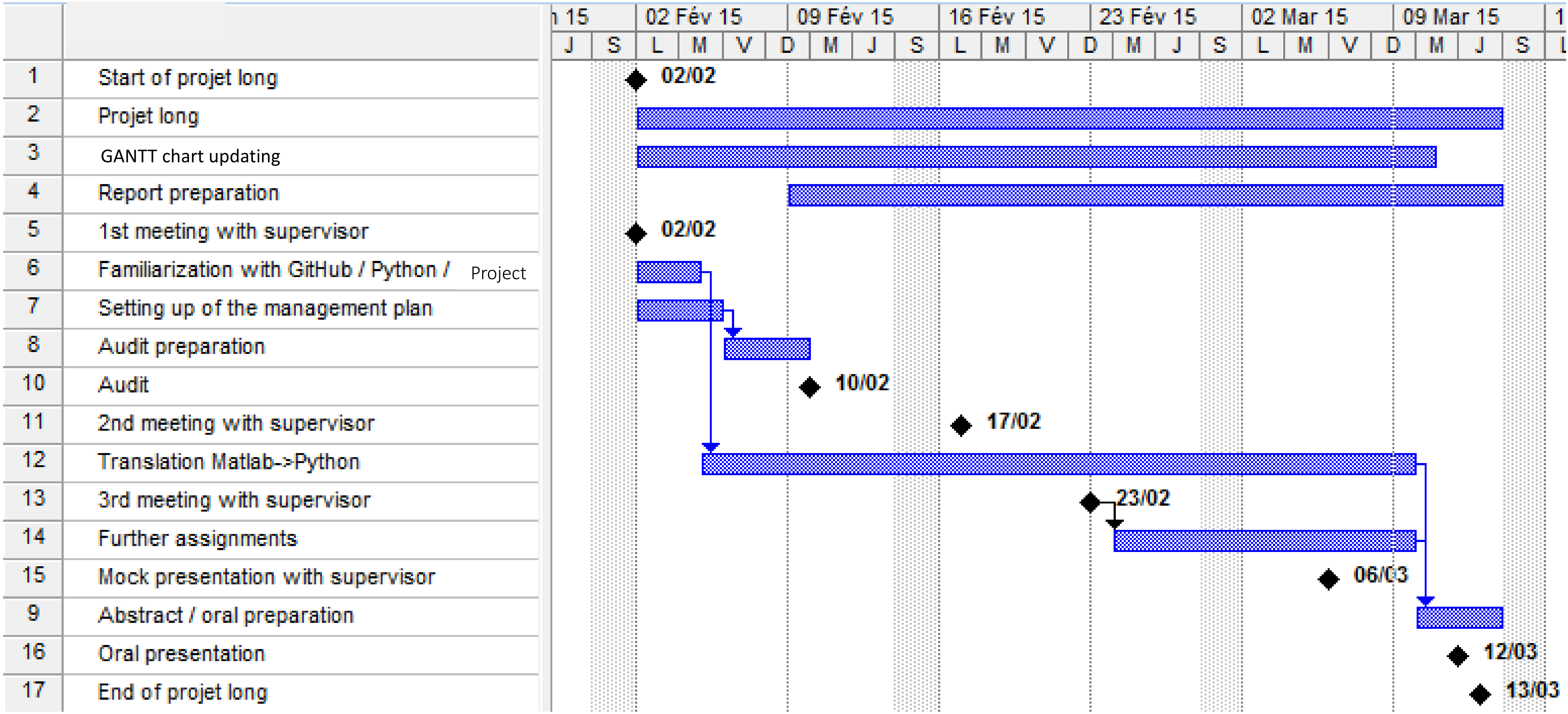
- Endangered species:
  - Leatherback turtles
  - Elephant seals

- ARGOS system → monitoring threatened species
- Creation of marine protected areas
- Matlab → Python

Contents

- Project management
- Graphical User Interface (GUI)
- ARGOS system
- Data extraction and common format
- Data processing
- Conclusion

GANTT chart


1 Start of projet long
2 Projet long
3 GANTT chart updating
4 Report preparation
5 1st meeting with supervisor
6 Familiarization with GitHub / Python / Project
7 Setting up of the management plan
8 Audit preparation
10 Audit
11 2nd meeting with supervisor
12 Translation Matlab->Python
13 3rd meeting with supervisor
14 Further assignments
15 Mock presentation with supervisor
9 Abstract / oral preparation
16 Oral presentation
17 End of projet long
02 Fév 15
09 Fév 15
16 Fév 15
23 Fév 15
02 Mar 15
09 Mar 15
02/02
02/02
10/02
17/02
23/02
06/03
12/03
13/03

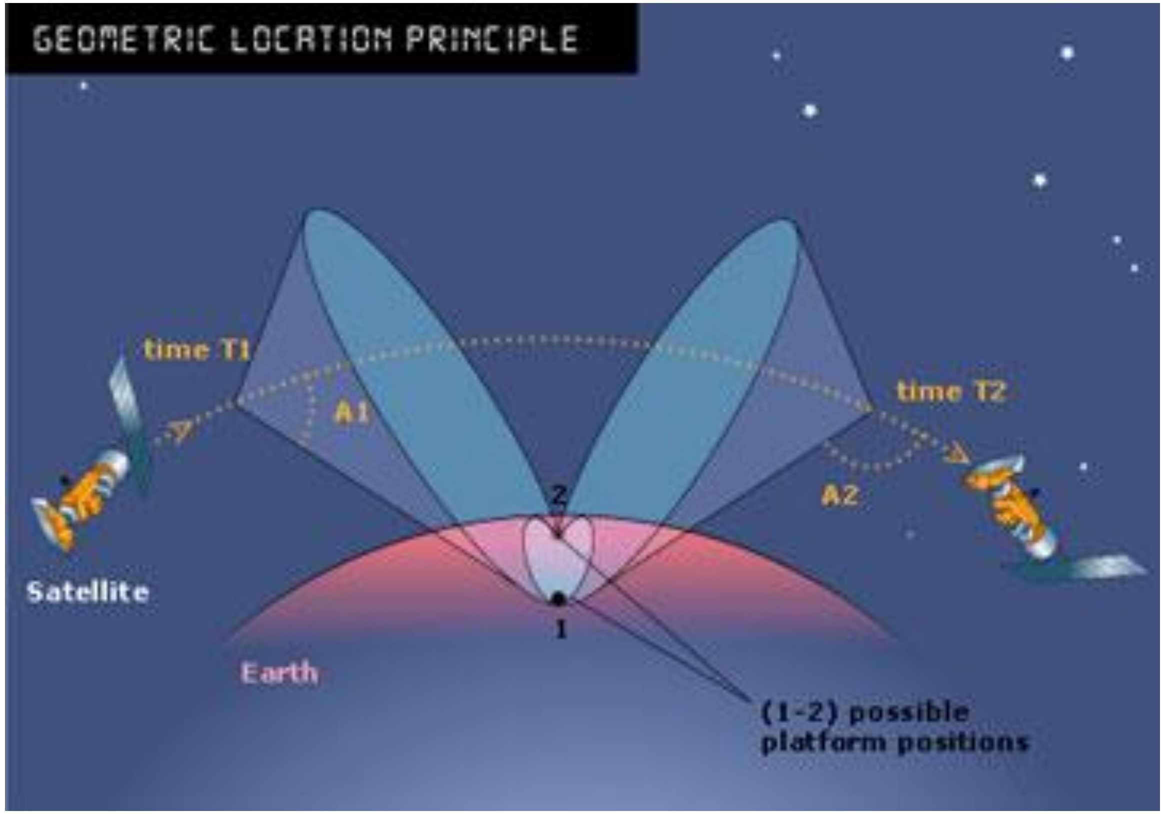
Work breakdown structure

Smoothing methods for ARGOS trajectories

- **Definition of the customer's expectations — WP1**
  - Overall understanding of the project — 1.1
  - Studying of the previous codes/work — 1.2
- **Familiarization with GitHub / Python / Project — WP2**
  - Introduction to GitHub — 2.1
  - Introduction to Python — 2.2
- **Setting up of a management plan — WP3**
  - GANTT chart — 3.1
  - WBS — 3.2
- **Tranlation Matlab / Python — WP4**
  - Developement — 4.1
  - test/debug — 4.2
  - Validation — 4.3
- **Further improvements — WP5**
  - Kalman smoothing — 5.1
  - Graphical User Interface — 5.2
- **Presentation of the project — WP6**
  - Summary of the work achieved — 6.1
  - Extended abstract — 6.2
  - Report — 6.3
  - Final oral presentation — 6.4

Graphical User Interface (GUI)

ARGOS system



Data extraction and common format

- ARGOS data stored in three different file formats :
  - CSV
  - DIAG
  - DS
- One program to rule them all, one program to find them, one program to bring them all and in the format bind them
- The remaining format is a list of dictionary

$$\left[\ dico[1] \quad dico[2[\quad \ldots \quad dico[n]\ \right]$$

Data extraction and common format

- Each transmission data are stored in a dictionary with a unique keys structure

$$\left\{ \begin{array}{c} "date" \\ "LC" \\ "lat" \\ "lon" \\ "lat\_image" \\ "lon\_image" \end{array} \right\}$$

- Key "date" associates to another dictionary

$$\left\{ \begin{array}{c} "annee" \\ "mois" \\ "jour" \\ "heure" \\ "min" \\ "sec" \end{array} \right\}$$

Data extraction and common format

- XML files contain parameters for smoothing methods and are specific to each species

- These parameters are also stored in a dictionary following XML reading.

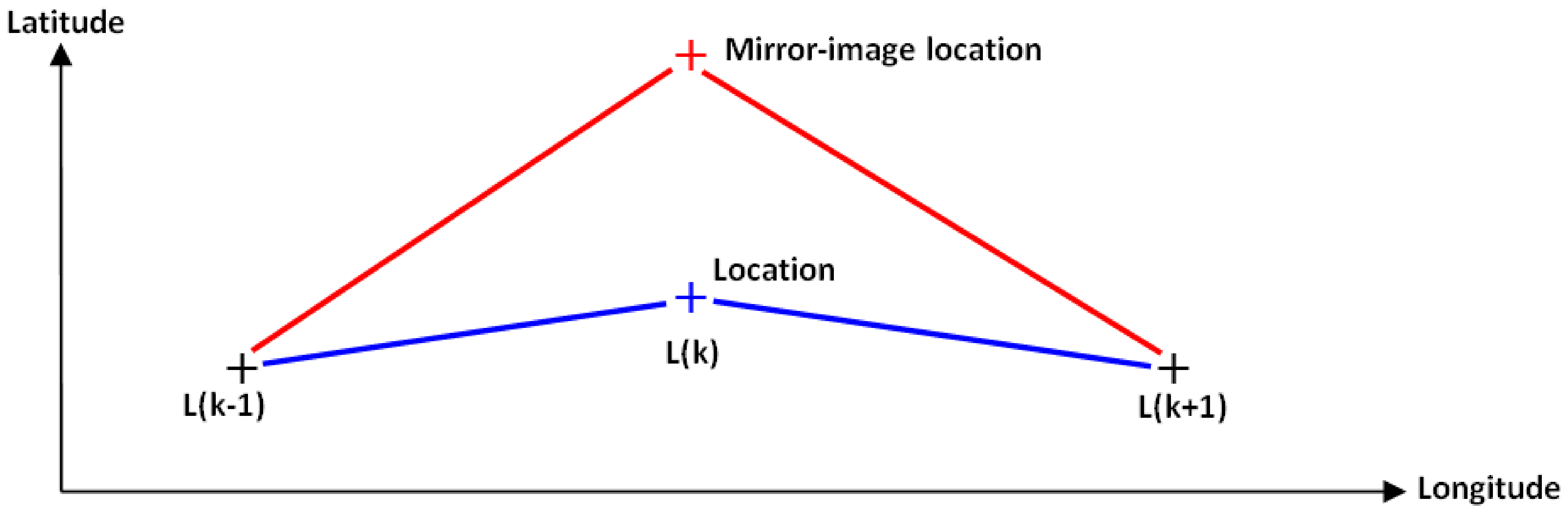
```xml
<?xml version="1.0" encoding="utf-8"?>

<!-- Fichier avec les parametres par defaut -->
<parametres>
    <lovi>
    <!-- Vitesse maximal de la tortue  en m/s-->
        <vitesse_max>2.8</vitesse_max>
    </lovi>
    <epan>
        <!-- pourcentage de points a conserver -->
        <ecart_max_pourcentage>95</ecart_max_pourcentage>
        <!-- intervale en secondes pour l'estimation -->
        <periode>28800</periode>
        <!-- nombre minimal de points dans la fenetre pou l'estimation 1  -->
        <min_estim1>5</min_estim1>
        <!-- nombre minimal de points dans la fenetre pou l'estimation 2  -->
        <min_estim2>5</min_estim2>
        <!-- taille minimale de la demifenetre pour l'estimation 1 -->
        <demi_fenetre_min_estim1>43200</demi_fenetre_min_estim1>
        <!-- taille maximale de la demifenetre pour l'estimation 1 -->
        <demi_fenetre_max_estim1>86400</demi_fenetre_max_estim1>
        <!-- taille minimale de la demifenetre pourl'estimation 2 -->
        <demi_fenetre_min_estim2>86400</demi_fenetre_min_estim2>
        <!-- taille maximale de la demifenetre pour l'estimation 2 -->
        <demi_fenetre_max_estim2>86400</demi_fenetre_max_estim2>
        <!-- nombre miniale de points pour la demifenetre pour l'estimation 1 -->
        <nb_pt_demi_fenetre_estim1>2</nb_pt_demi_fenetre_estim1>
        <!-- nombre miniale de points pour la demifenetre pour l'estimation 2 -->
        <nb_pt_demi_fenetre_estim2>2</nb_pt_demi_fenetre_estim2>
    </epan>
</parametres>
```

Data processing

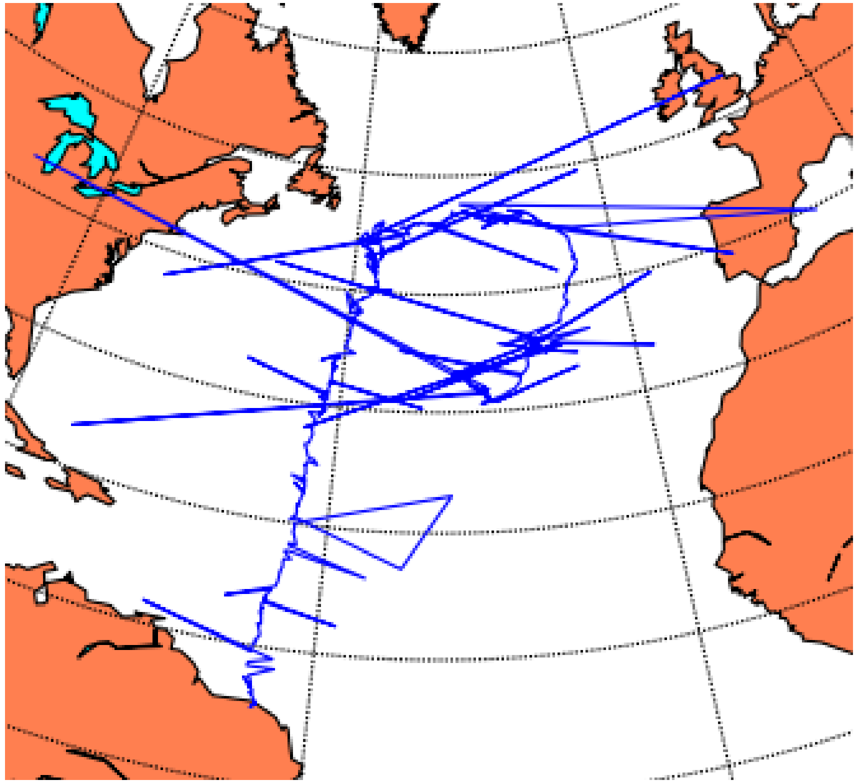
Preprocessing

CHOICE OF LOCATION

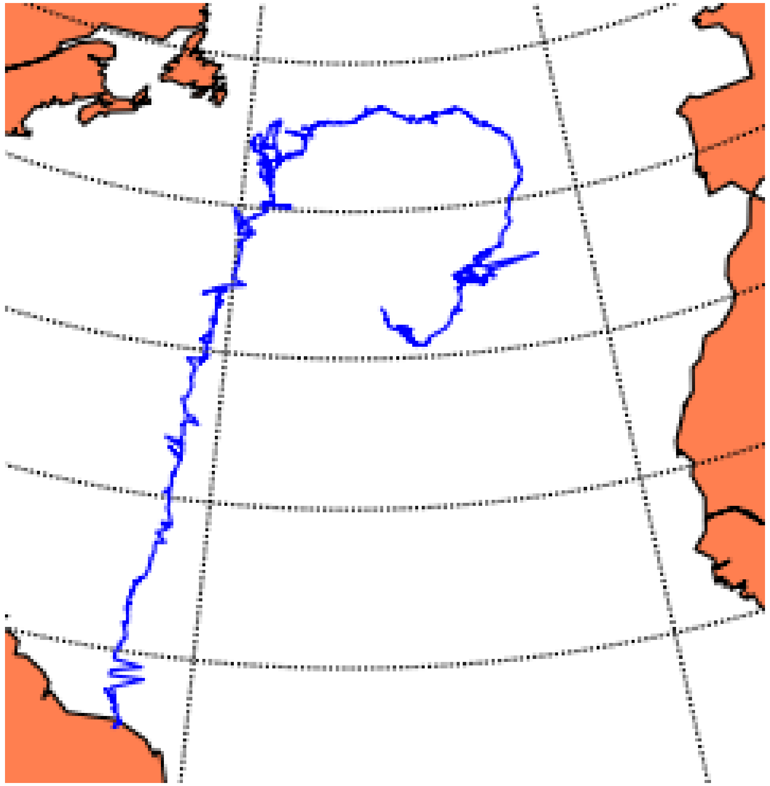
Choice of location



Choice of location



Raw data



Data preprocessed with correction of location

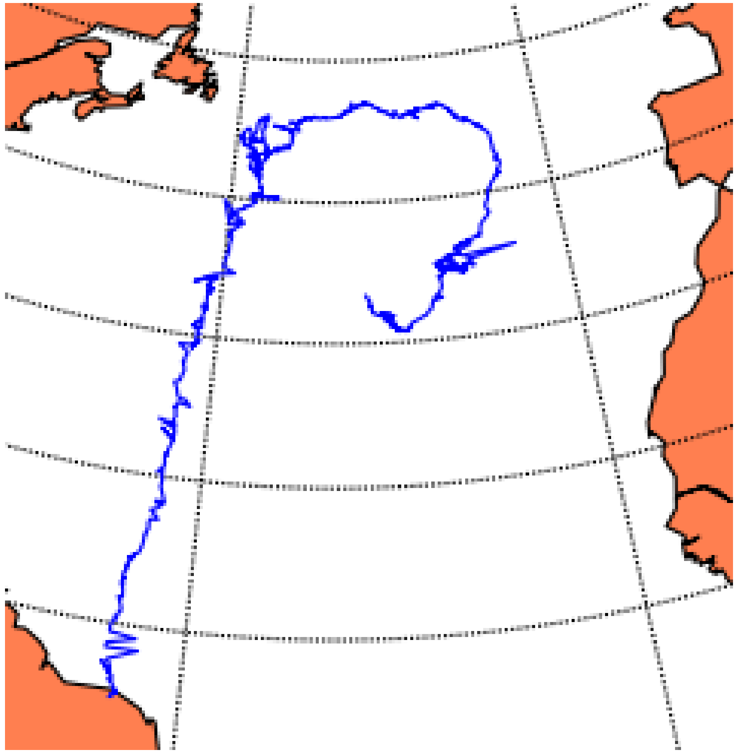
Preprocessing

DELETION OF EXCESSIVE SPEED

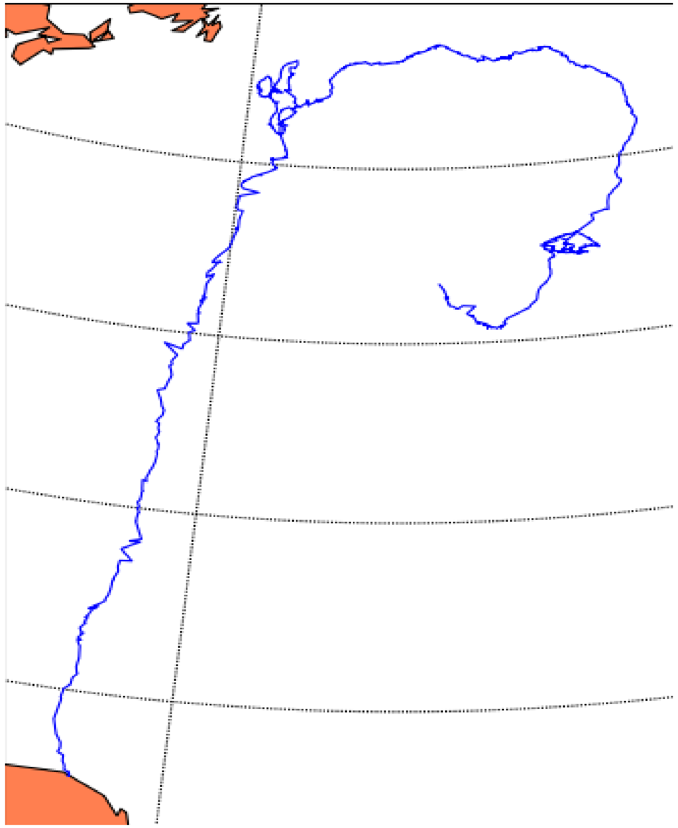
Deletion of excessive speed

- Computation of speed between two points
- Criteria of precision of the location
- Comparison with the specie’s maximal speed

Deletion of excessive speed



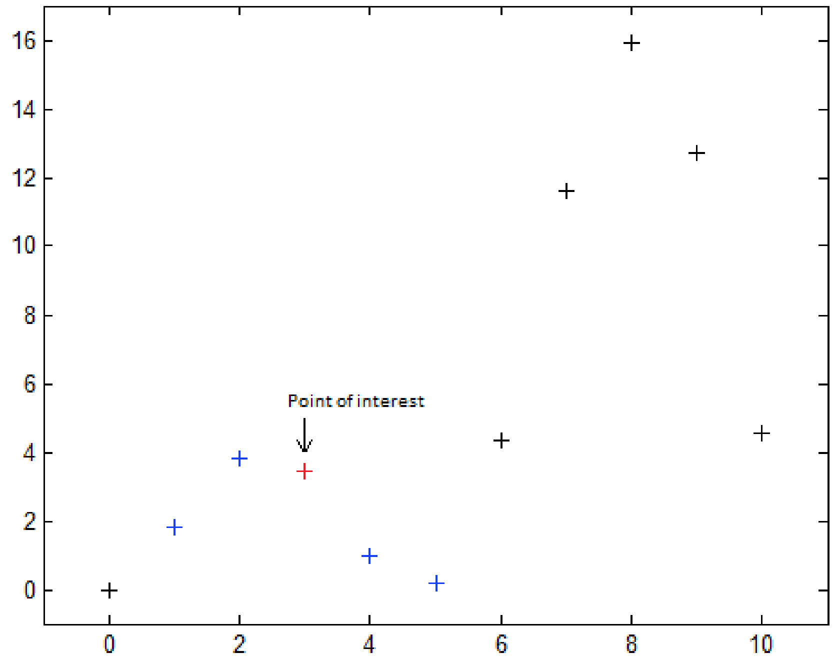
Data before deletion of excessive speed



Data after deletion of excessive speed

Data processing

- Here you estimate one position as the weighted sum of the two previous, current and two following positions



2 different weights :

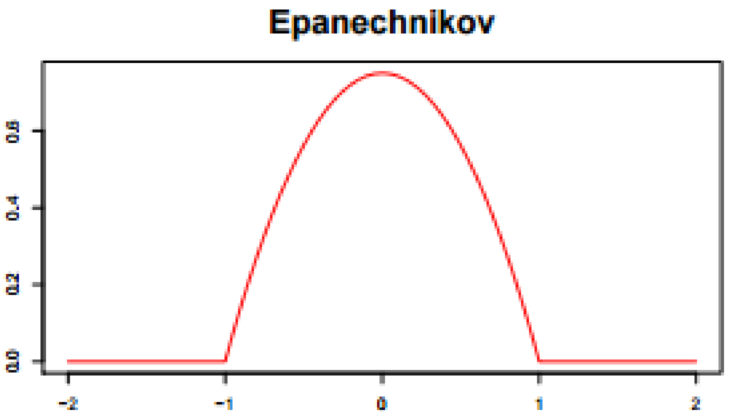
- one from the kernel
- one from the quality of the ARGOS localization

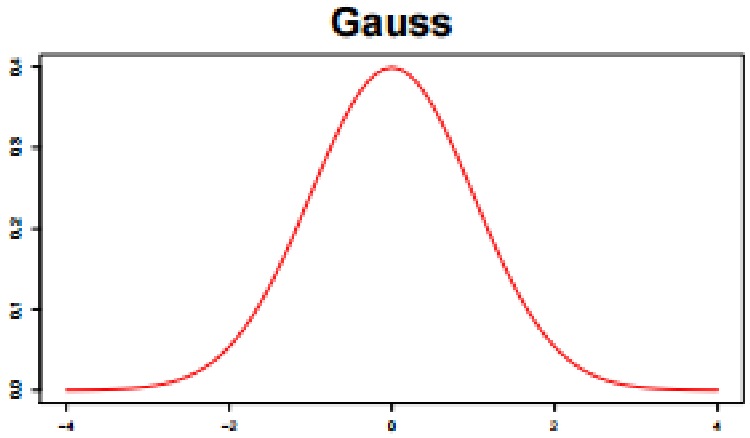
Data processing

- Adaptable size of the support of the epanechnikov kernel :

$$\frac{3}{4h}.\left(1-\frac{x^2}{h}\right)$$ with 2h = size of the support

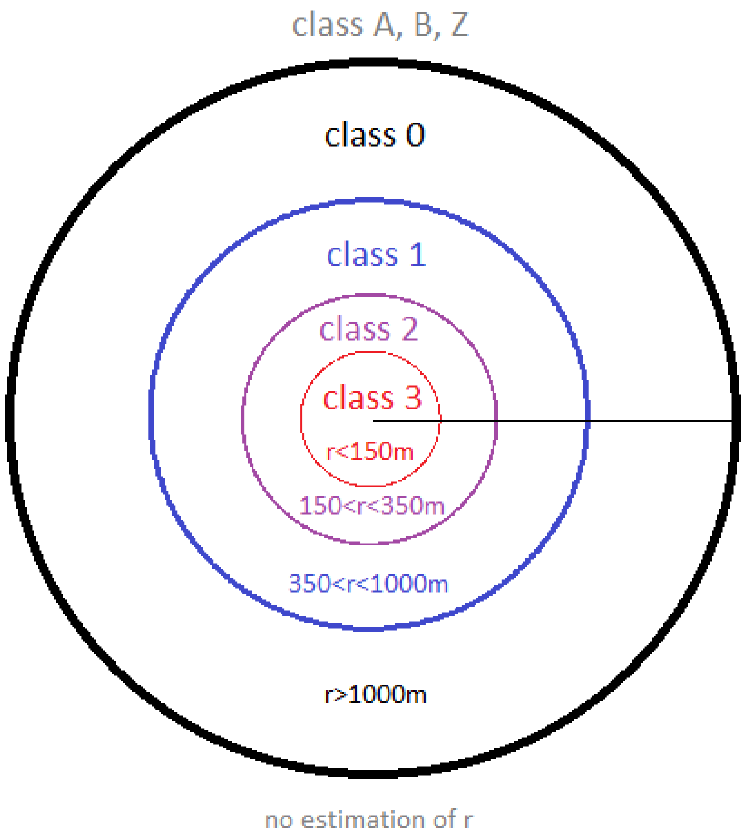
- Epanechnikov kernel minimizes AMISE (Asymptotic Mean Integrated Squared Error) and is therefore optimal.





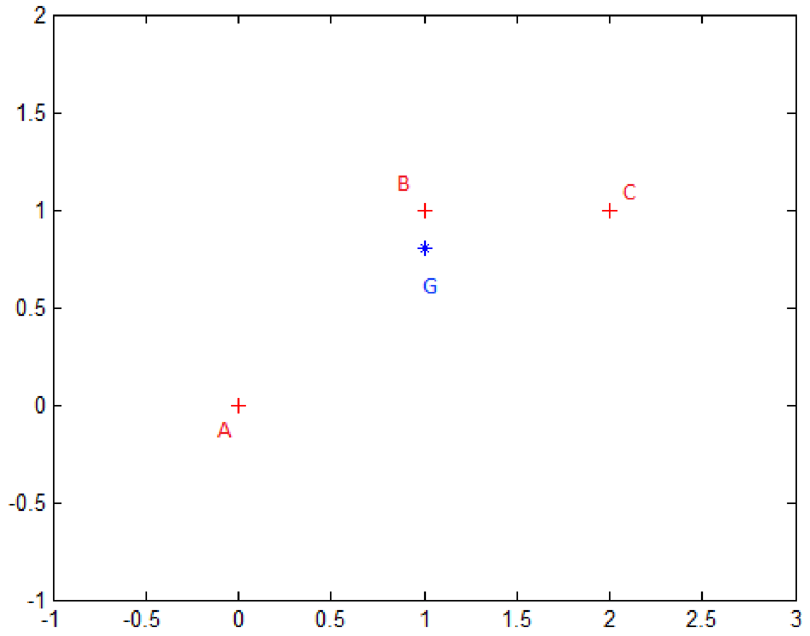
Data processing

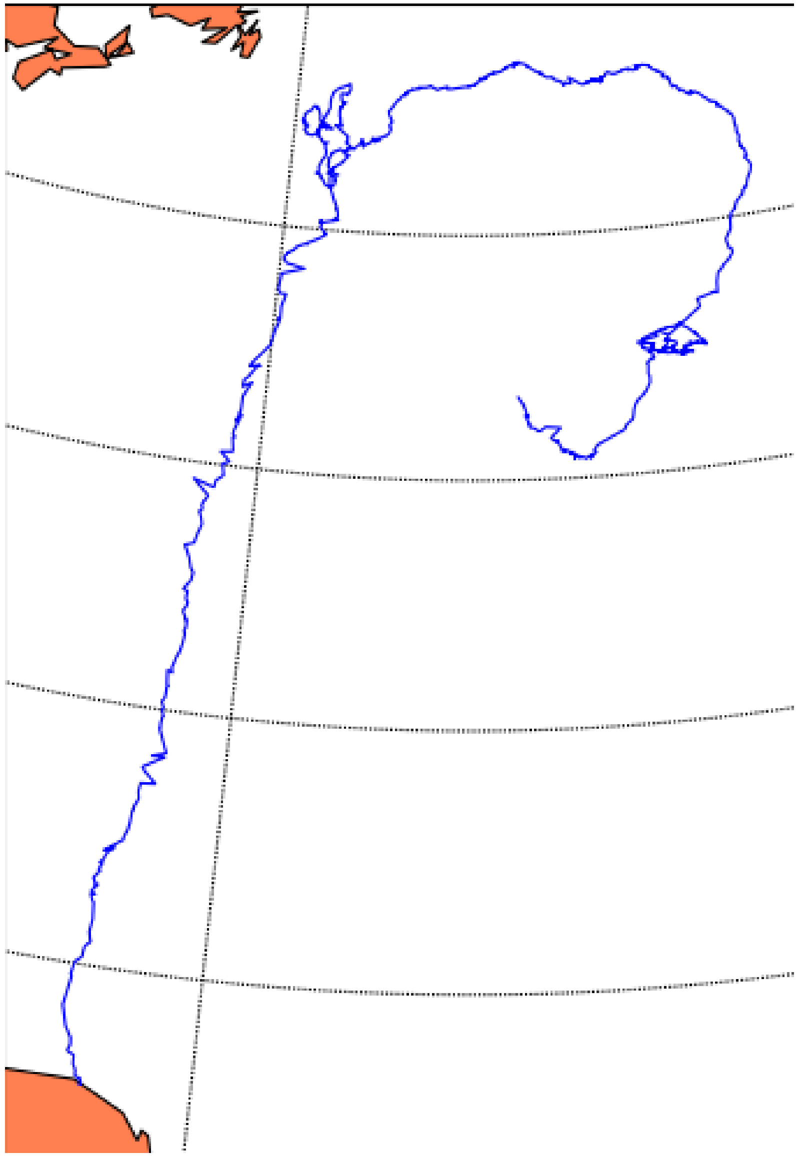
- The weights increase as the precision of measurement increases



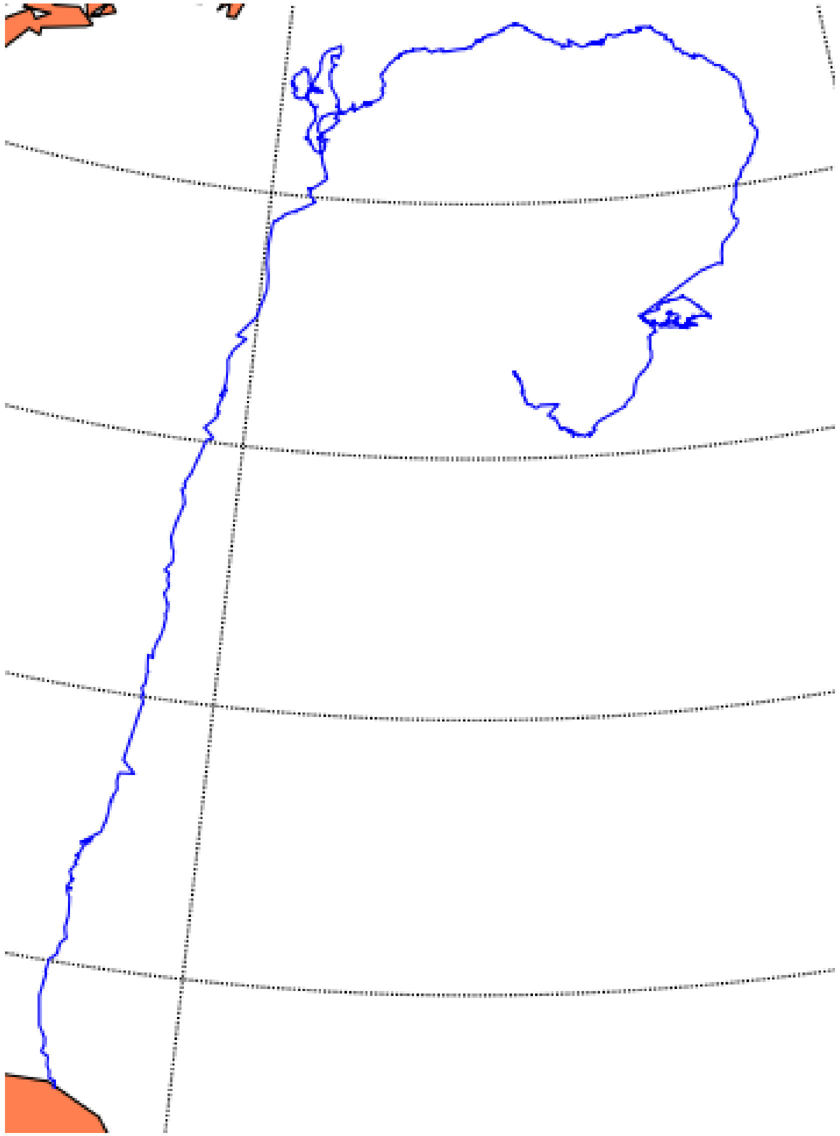
Data processing

If the estimated position is too far from the ARGOS position, this position is removed

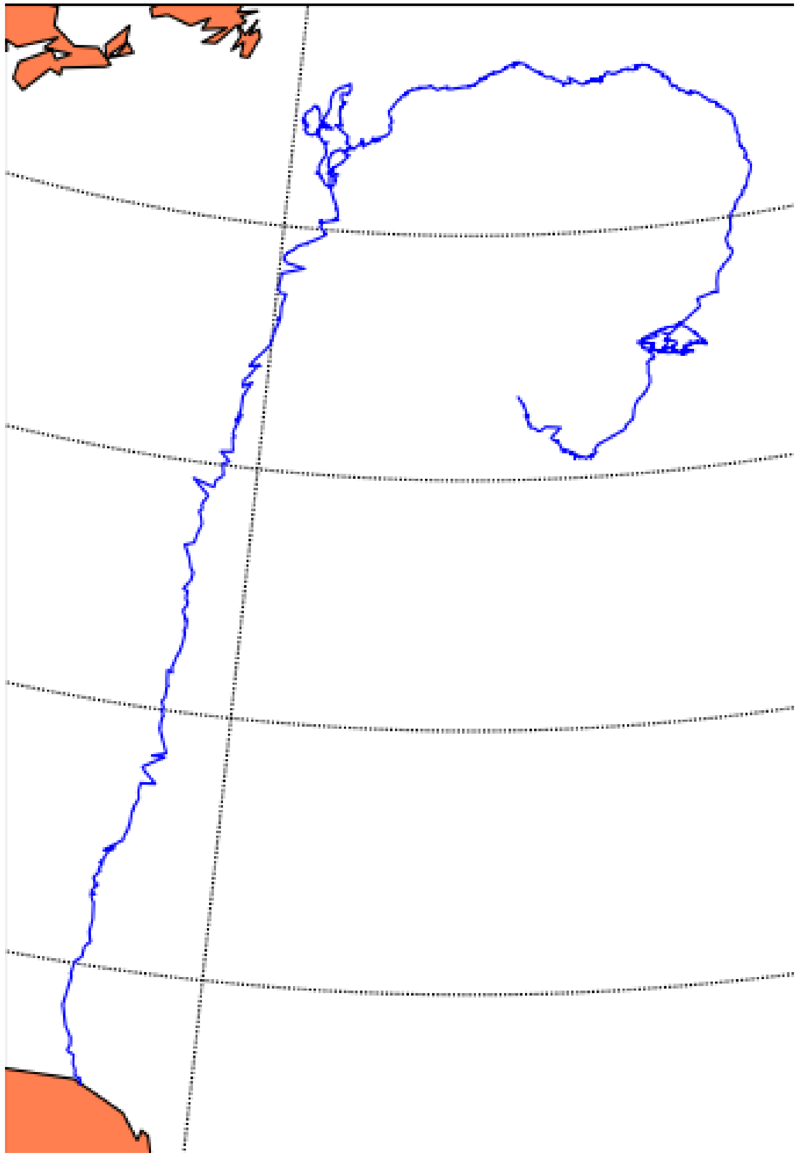




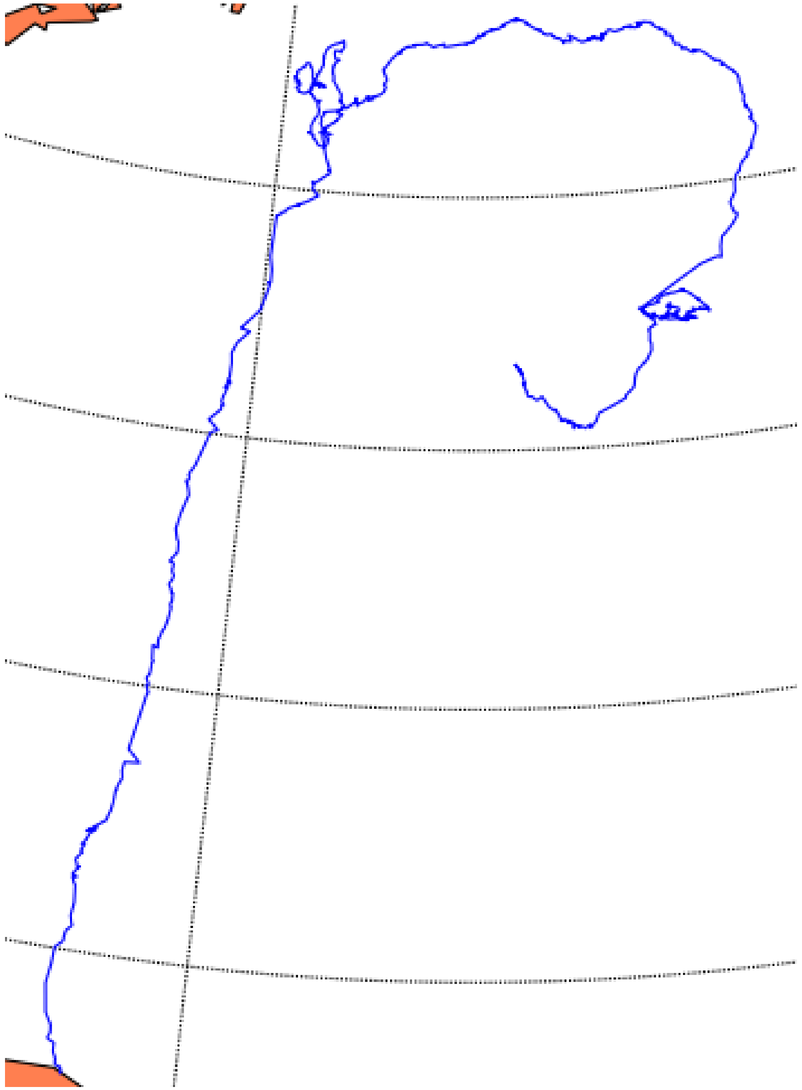
Trajectory before the estimation



Trajectory with an Epanechnikov kernel



Trajectory before the estimation

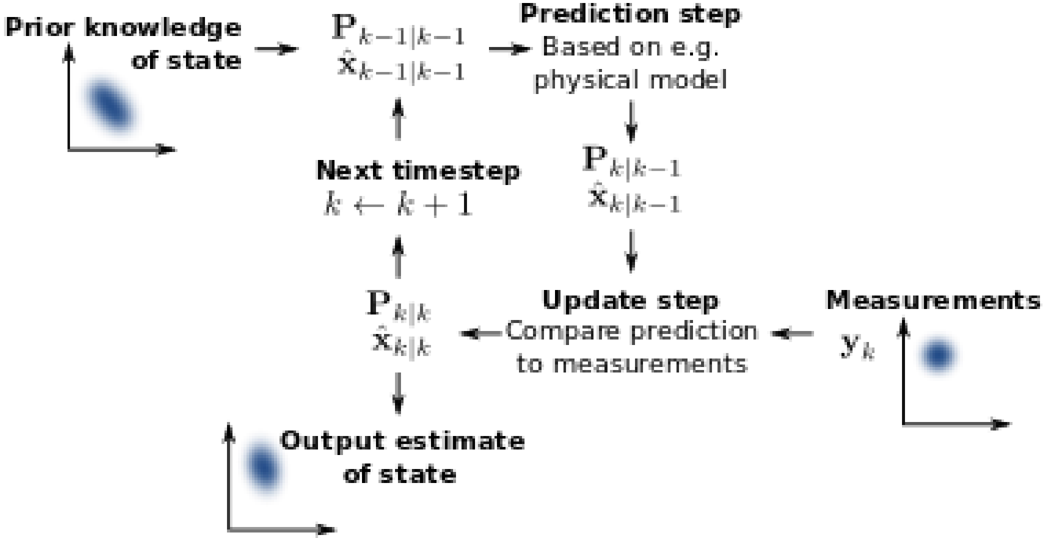


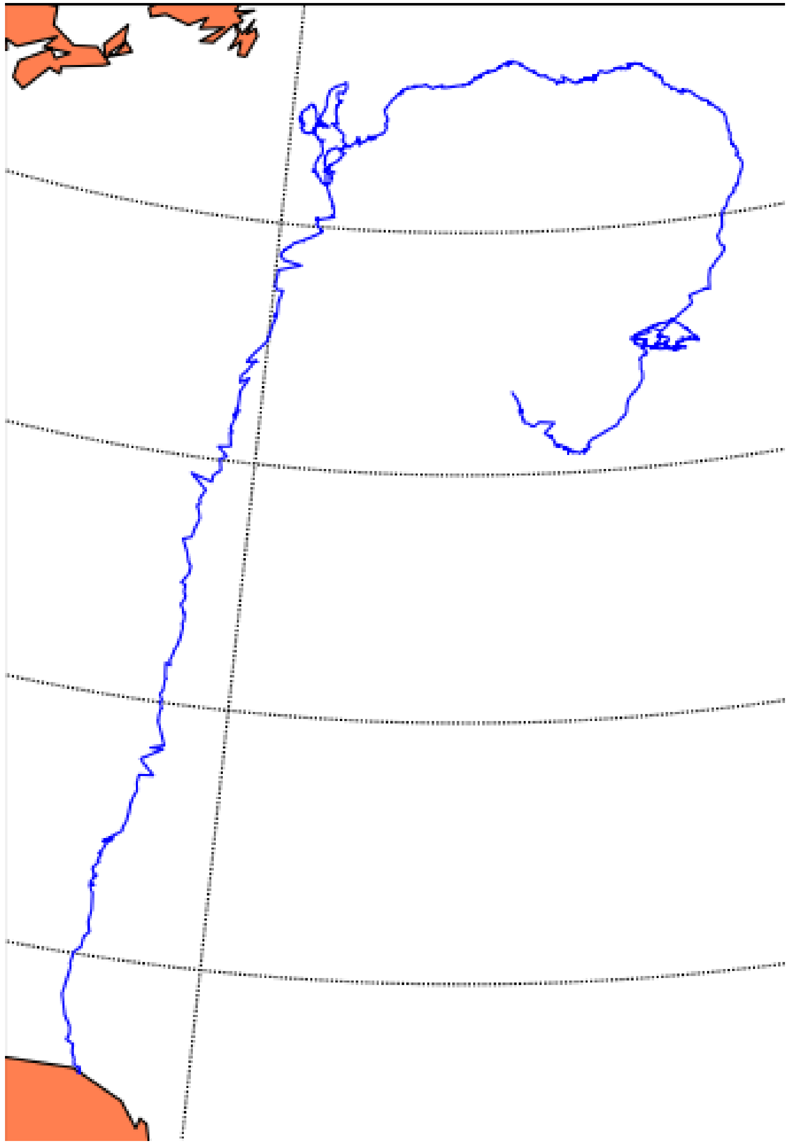
Trajectory with a Gaussian kernel

Kalman smoothing

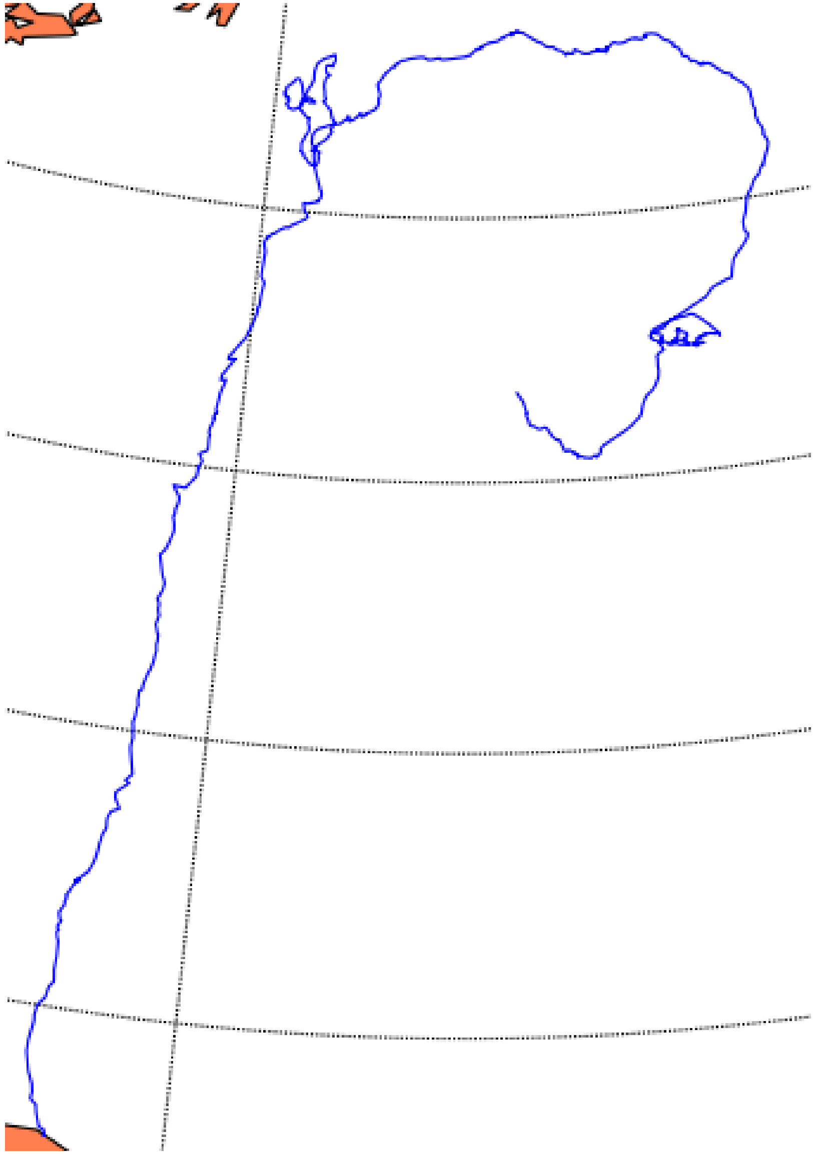
- An expectation maximization algorithm estimates the transition matrix

- Use of all the data in order to smooth the trajectories





Trajectory before Kalman smoothing

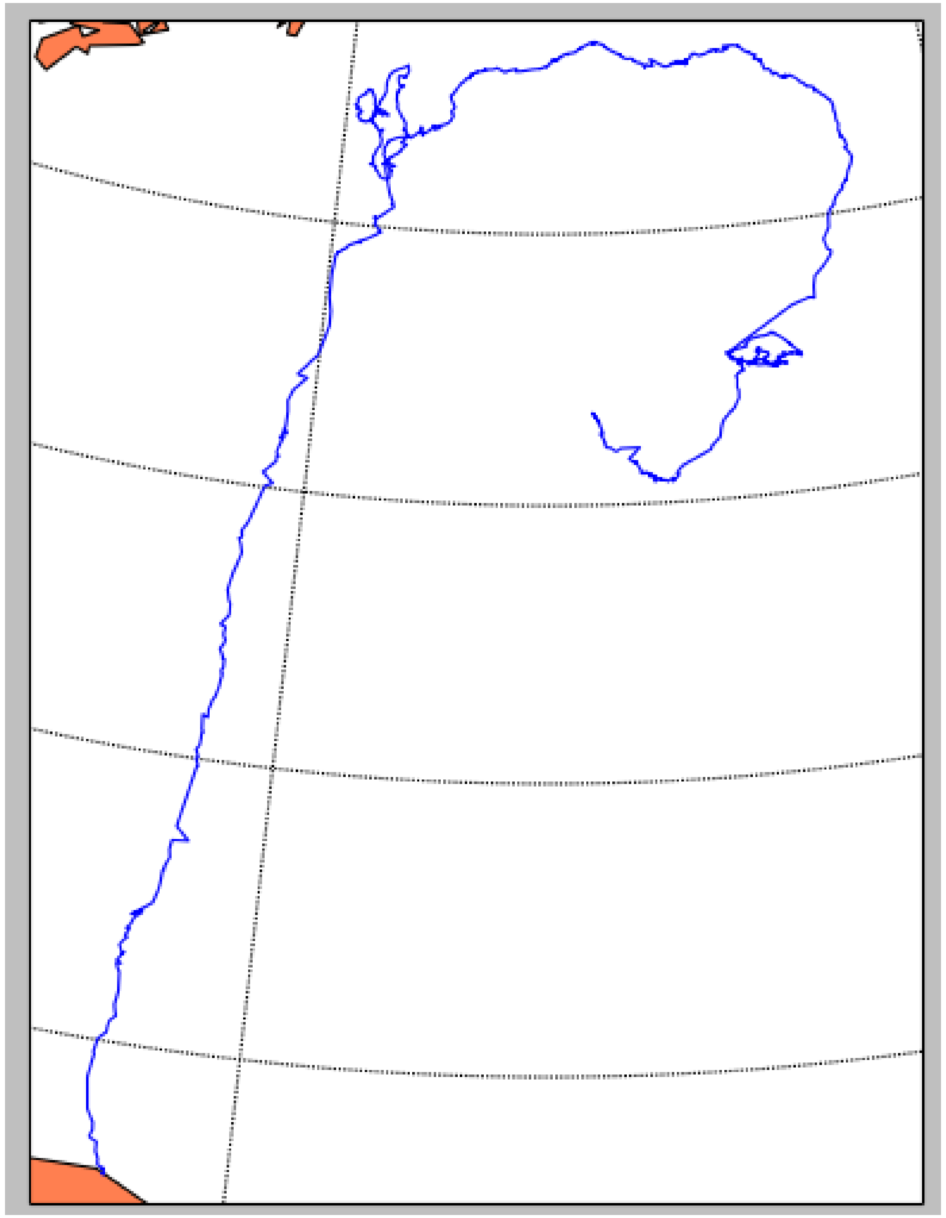


Trajectory after Kalman smoothing

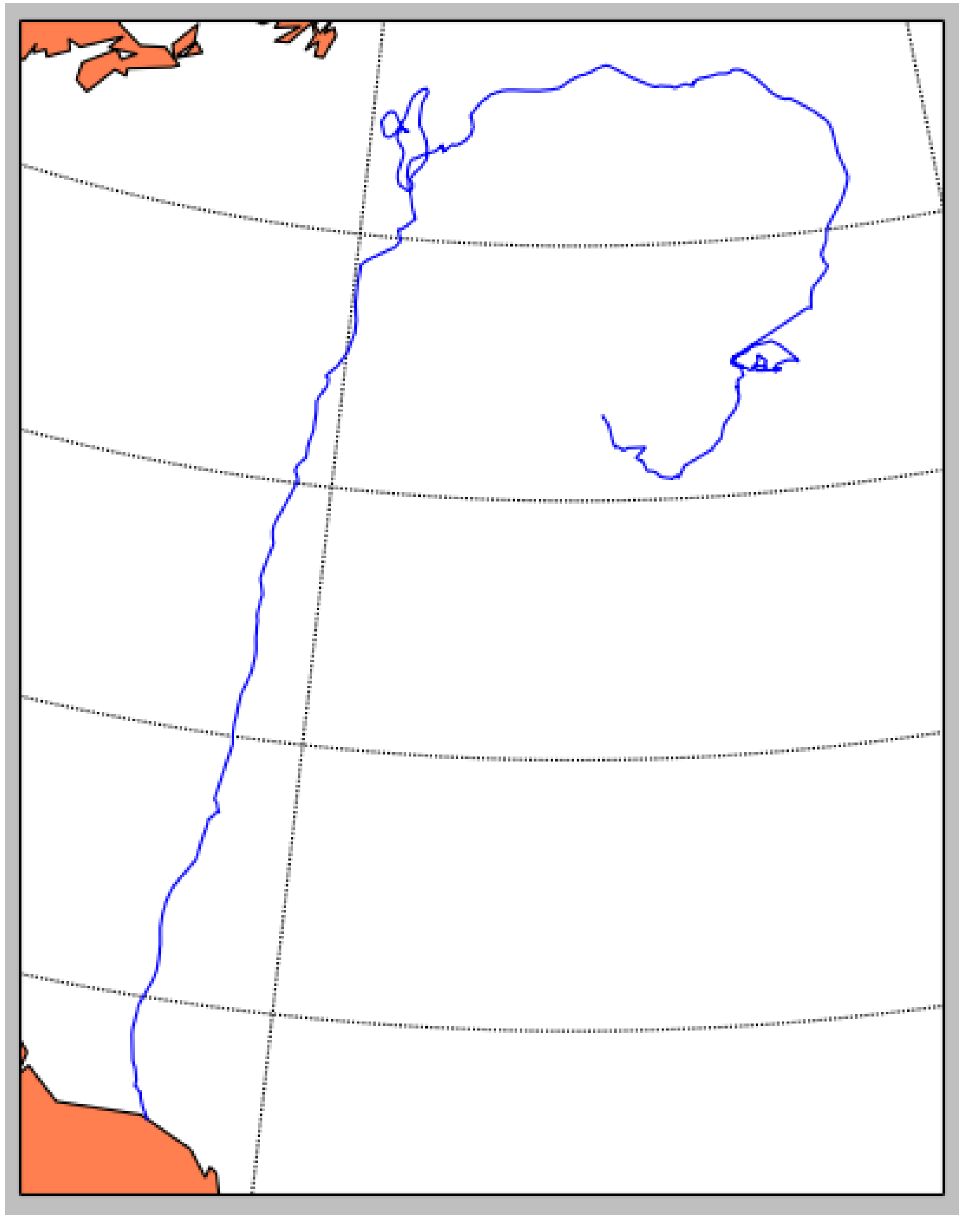
Second estimation

- Linear regression
- Resampling trajectory

➡ Points spaced with a constant time step size



Output without the second estimation



Output with the second estimation

Conclusion

- Efficient and reliable algorithms
- Work achieved intends to monitor endangered species
- Trajectories can be plotted and exploited using the GUI
- Further improvements:
  - Comparison with GPS data
  - Handle new ARGOS data