Animating a sequence of maps:

Design document for integrating animal telemetry data with oceanographic data: using downloaded data files

Project for Northeastern University GIS 6345 (Fall 2025)

Steve Lockhart

Contents

Introduction

For the GIS 6345 project, I used this document to keep track of design decisions, data sources, issues/resolutions, references, so I don't have to go hunting for that information later. The project deliverable (showing plots, etc.) will be a powerpoint presentation (not ready yet).

In this project for GIS 6345 at Northeastern University, I accomplished the following:

- Used animation to show how a set of marine mammal tracks evolve relative to a time-varying oceanographic feature, the meandering Gulf Stream.
- For each marine mammal track, calculated its distance from a specified location as a function of time.

For the marine mammal tracks:

- I downloaded a dataset from movebank.org.

For the oceanographic data, I have two scenarios:

1) The oceanographic data (sea surface temperature and current velocity) is from HYCOM netcdf files.
2) The oceanographic data (sea surface temperature) is from MUR SST analysis data.

This project involves the following aspects of python geospatial programming:

- Conversion of animal telemetry data from csv to pandas dataframe to numpy array
- Conversion of HYCOM oceanographic data from netcdf to pandas dataframe to numpy array
- Conversion of MUR SST oceanographic data from csv to pandas dataframe to numpy array
- Integrating vector data (animal track) with raster data (oceanographic data)
- Animating maps of the integrated data sets using python packages matplotlib and celluloid
- Conversion from one coordinate system (with decimal degrees as units) to another (with meters as units), so I can calculate distance. I use the geopandas .to_crs() method to convert from WGS84 to the appropriate UTM.

An alternative approach—using Google Earth Engine (instead of downloading the data files)—will be covered in a separate document.

Approach

To animate a sequence of maps, I considered the following approaches:

- In the reading materials for Assignment 11, I saw this link at https://medium.com/udacity/creating-map-animations-with-python-97e24040f17b where the author says:

  > "My general strategy was to generate an image for each day in the dataset, then convert all those images into a video. To do this, I used Python and a few packages: Pandas, for loading and manipulating the data, Cartopy, for drawing the map, and Matplotlib, for plotting the data. After generating all the images, I used ffmpeg to combine the individual frames into the video..."

- Matplotlib FuncAnimation? Also see Matplotlib demos at https://matplotlib.org/stable/gallery/

- Matplotlib and celluloid

- Plotly with dash. See https://plotly.com/python/animations/

  "Although Plotly Express supports animation for many chart and map types, smooth inter-frame transitions are today only possible for scatter and bar"

(I started with Matplotlib animation. It worked fine for a simple line plot e.g. animating a point along a track. However, when I tried to add animation of the overlays e.g. the contour plot of sea-surface temperature, I ran into problems. With a little research, I found it easier to animate Matplotlib figures using the celluloid package.)

Data view

Input data

Animal telemetry data

After searching several sites[1], I found the animal telemetry data for a study named "Short-finned pilot whales CRC NW Atlantic" on movebank.org at:

https://www.movebank.org/cms/webapp?gwt_fragment=page%3Dstudies%2Cpath%3Dstudy31575534

As documented in Thorne et al. (2017), there are a set of marine mammal tracks from this dataset that fall within a time window from 10/15/2015 to 01/05/2016. These tracks cover the region of interest-- longitude range (in degrees) of [-75.0 -73.0] and latitude range of [34.25 37.5].

For this project, I analyzed the tracks for the following tad IDs:

- GmTag137
- GmTag142

Case 1: Environmental data from HYCOM model hindcasts

In this case, the oceanographic data was obtained from HYCOM model hindcasts, stored in netcdf files. There is a netcdf file every 3 hours, and the variables of interest include the following:

- temperature (T)
- 2 components of horizontal velocity (u,v)

For a given netcdf file, these variables are stored as a function of longitude, latitude, and depth.

To match the time period of the animal tracks, I had to use an older (lower-resolution) set of HYCOM model hindcasts. (See Issues section.) This "GLBv0.08 experiment 56.3" version of the HYCOM ocean model has a spatial resolution of .08 degree in latitude and .08 degree in longitude.

From my job, I have Matlab code that downloads a set of netcdf files from the HYCOM ftp server and trims them to the desired region of interest (a bounding box in longitude, latitude). So, to get the data for the selected animal tracks, I downloaded one HYCOM file per day from

[1] See SLockhart_Explore_Animal_Telemetry_Data.docx

10/15/2015 to 01/06/2016 and trimmed them to the region of interest. See the “Code prerequisites” section below.

For this older version of HYCOM, the ftp server is at:

https://data.hycom.org/datasets/GLBv0.08/expt_56.3/data/2015/ or

https://data.hycom.org/datasets/GLBv0.08/expt_56.3/data/2016/

Case 2: Environmental data from MUR SST analysis data

In this case, the sea surface temperature (SST) data was obtained from the MUR SST analysis model at:

https://coastwatch.pfeg.noaa.gov/erddap/griddap/jplMURSST41F.html

I chose a time window (10/15/2015 to 01/06/2016) and region of interest (longitude range (in degrees) of [-75.0 -73.0] and latitude range of [34.25 37.5]), and downloaded a csv file.

The temporal resolution of this data set is daily i.e. an array of sea-surface temperature per day. However, the csv file itself contains multiple days.

The spatial resolution of MUR SST is 0.01 degree. However, I reduced it to .04 degrees due to performance issues on my laptop. (See issues.)

Process view

This section documents the inputs and outputs for each process.

Animation of animal track with HYCOM data

Input(s)	• Animal tracks from csv • SST and surface current from HYCOM netcdf files (1 file per day)
Processing	• With env_data_src set to 'HYCOM', **project_main.py** calls functions in **project_functions.py**
Output(s)	• I animated two animal tracks. Each animated track (with overlays of time-varying oceanographic data) is stored in a mp4 file, posted on youtube at: https://youtu.be/0TiQmDO2qME https://youtu.be/5yLqj9c-eOc

Animation of animal track with MUR SST data

Input(s)	• Animal tracks from csv • SST from MUR SST csv, containing a SST array per day for many days
Processing	• **project_pickle_MUR_SST.py** calls function in **project_functions.py** • With env_data_src set to 'MUR', **project_main.py** calls functions in **project_functions.py**
Output(s)	• I animated two animal tracks. Each animated track (with overlays of time-varying oceanographic data) is stored in a mp4 file, posted on youtube at: https://youtu.be/ZY_taGq_H8c https://youtu.be/lEMXJDIT0kE

Distance vs. time for animal tracks

Input(s)	• Animal tracks from csv
Processing	• **project_plot_distance.py** calls function in **project_functions.py**
Output(s)	• Plot

Deployment view

Code prerequisites

Dependency	***Comments***
Matlab code	• To download and trim the HYCOM netcdf files to the region of interest, I'm reusing some Matlab code from my work. Note that this Matlab code is not included in the project's github repository.

Code dependencies

Dependency	***Comments***
ffmpeg	• Downloaded from https://github.com/BtbN/FFmpeg-Builds/releases for my platform (Windows 10) • Needed for Matplotlib animation to create an mp4 file[2]

Code repository

The python code is stored in a github repository at:
https://github.com/sblockhartzzero/GIS6345

Python environment

Using conda, I created a new virtual environment gis6345_project1, activated it, and installed the following packages in the new env (in the following order):

- **rasterio**
- **numpy (included in above)**
- **xarray**
- **pandas (included in above)**
- **netCDF4**
- **matplotlib**

[2] Subsequently, I switched to using celluloid, so I'm not sure it's still using ffmpeg to generate the mp4.

- datetime –failed to solve environment, but it seems like I already have it (from base I guess) e.g. I can import it…
- **celluloid**
- **geopandas**
- **shapely (included in above)**

When I installed this package (using conda)…	***…it also installed these packages***
rasterio	The following packages will be downloaded: affine-2.4.0 fontconfig-2.15.0 libnetcdf-4.9.3 libpng-1.6.50 libpq-17.6 mkl-service-2.5.2 numpy-2.3.3 numpy-base-2.3.3 pip-25.2 poppler-data-0.4.12 python-3.13.9 rasterio-1.4.3 setuptools-80.9.0 The following NEW packages will be INSTALLED: affine pkgs/main/win-64::affine-2.4.0-py313haa95532_0 attrs pkgs/main/win-64::attrs-24.3.0-py313haa95532_0 blas pkgs/main/win-64::blas-1.0-mkl blosc pkgs/main/win-64::blosc-1.21.6-h4190f5b_0 bzip2 pkgs/main/win-64::bzip2-1.0.8-h2bbff1b_6 ca-certificates pkgs/main/win-64::ca-certificates-2025.9.9-haa95532_0 cairo pkgs/main/win-64::cairo-1.18.4-he9e932c_0 certifi pkgs/main/win-64::certifi-2025.10.5-py313haa95532_0 cfitsio pkgs/main/win-64::cfitsio-3.470-h2bbff1b_7 click pkgs/main/win-64::click-8.2.1-py313haa95532_0 click-plugins conda-forge/noarch::click-plugins-1.1.1.2-pyhd8ed1ab_0 cligj pkgs/main/noarch::cligj-0.7.2-pyhd3eb1b0_0 colorama pkgs/main/win-64::colorama-0.4.6-py313haa95532_0 expat pkgs/main/win-64::expat-2.7.1-h8ddb27b_0 fontconfig pkgs/main/win-64::fontconfig-2.15.0-hd211d86_0 freeglut pkgs/main/win-64::freeglut-3.4.0-h8a1e904_1 freetype pkgs/main/win-64::freetype-2.13.3-h0620614_0 freexl pkgs/main/win-64::freexl-2.0.0-hd7a5696_0 geos pkgs/main/win-64::geos-3.10.6-he74ecf9_0 geotiff pkgs/main/win-64::geotiff-1.7.0-hbf7d8e7_4 hdf4 pkgs/main/win-64::hdf4-4.2.13-h712560f_2

	hdf5 pkgs/main/win-64::hdf5-1.14.5-ha36df97_2 icc_rt pkgs/main/win-64::icc_rt-2022.1.0-h6049295_2 intel-openmp pkgs/main/win-64::intel-openmp-2025.0.0-haa95532_1164 jpeg pkgs/main/win-64::jpeg-9f-ha349fce_0 kealib pkgs/main/win-64::kealib-1.5.0-h6c4040a_2 lcms2 pkgs/main/win-64::lcms2-2.16-h62be587_1 lerc pkgs/main/win-64::lerc-4.0.0-h5da7b33_0 libaec pkgs/main/win-64::libaec-1.1.3-hcdb6601_0 libboost pkgs/main/win-64::libboost-1.88.0-h3d2b97d_0 libcurl pkgs/main/win-64::libcurl-8.15.0-h2300eb9_0 libdeflate pkgs/main/win-64::libdeflate-1.22-h5bf469e_0 libffi pkgs/main/win-64::libffi-3.4.4-hd77b12b_1 libgdal pkgs/main/win-64::libgdal-3.6.2-hdf1f3d4_10 libglib pkgs/main/win-64::libglib-2.84.2-h405b238_0 libiconv pkgs/main/win-64::libiconv-1.16-h2bbff1b_3 libkml pkgs/main/win-64::libkml-1.3.0-h63940dd_7 libkrb5 pkgs/main/win-64::libkrb5-1.21.3-h885b0b7_4 libmpdec pkgs/main/win-64::libmpdec-4.0.0-h827c3e9_0 libnetcdf pkgs/main/win-64::libnetcdf-4.9.3-h0c3980a_0 libpng pkgs/main/win-64::libpng-1.6.50-h46444df_0 libpq pkgs/main/win-64::libpq-17.6-h652a1e2_0 libspatialite pkgs/main/win-64::libspatialite-5.1.0-h6715bb4_2 libssh2 pkgs/main/win-64::libssh2-1.11.1-h2addb87_0 libtiff pkgs/main/win-64::libtiff-4.7.0-h404307b_0 libwebp-base pkgs/main/win-64::libwebp-base-1.3.2-h3d04722_1 libxml2 pkgs/main/win-64::libxml2-2.13.8-h866ff63_0 libzip pkgs/main/win-64::libzip-1.8.0-h289538f_1 libzlib pkgs/main/win-64::libzlib-1.3.1-h02ab6af_0 lz4-c pkgs/main/win-64::lz4-c-1.9.4-h2bbff1b_1 minizip pkgs/main/win-64::minizip-4.0.3-hb68bac4_0 mkl pkgs/main/win-64::mkl-2025.0.0-h5da7b33_930 mkl-service pkgs/main/win-64::mkl-service-2.5.2-py313h0b37514_0 mkl_fft pkgs/main/win-64::mkl_fft-1.3.11-py313h5810407_1 mkl_random pkgs/main/win-64::mkl_random-1.2.8-py313hedd7022_1 numpy pkgs/main/win-64::numpy-2.3.3-py313h050da96_1 numpy-base pkgs/main/win-64::numpy-base-2.3.3-py313h1e017a8_1 openjpeg pkgs/main/win-64::openjpeg-2.5.2-h9b5d1b5_1 openssl pkgs/main/win-64::openssl-3.0.18-h543e019_0 pcre2 pkgs/main/win-64::pcre2-10.42-h0ff8eda_1 pip pkgs/main/noarch::pip-25.2-pyhc872135_1 pixman pkgs/main/win-64::pixman-0.46.4-h4043f72_0 poppler pkgs/main/win-64::poppler-24.09.0-h10b5248_4 poppler-data pkgs/main/win-64::poppler-data-0.4.12-haa95532_0 proj pkgs/main/win-64::proj-9.3.1-h4c8f42b_1 pyparsing pkgs/main/win-64::pyparsing-3.2.0-py313haa95532_0 python pkgs/main/win-64::python-3.13.9-h260b955_100_cp313 python_abi pkgs/main/win-64::python_abi-3.13-1_cp313 qhull pkgs/main/win-64::qhull-2020.2-h59b6b97_2 rasterio pkgs/main/win-64::rasterio-1.4.3-py313he5c4563_0 setuptools pkgs/main/win-64::setuptools-80.9.0-py313haa95532_0 sqlite pkgs/main/win-64::sqlite-3.50.2-hda9a48d_1 tbb pkgs/main/win-64::tbb-2022.0.0-h214f63a_0 tbb-devel pkgs/main/win-64::tbb-devel-2022.0.0-h214f63a_0

	tiledb pkgs/main/win-64::tiledb-2.3.3-hd8964de_3 tk pkgs/main/win-64::tk-8.6.15-hf199647_0 tzdata pkgs/main/noarch::tzdata-2025b-h04d1e81_0 ucrt pkgs/main/win-64::ucrt-10.0.22621.0-haa95532_0 uriparser pkgs/main/win-64::uriparser-0.9.8-h1a0bd13_0 vc pkgs/main/win-64::vc-14.42-haa95532_5 vc14_runtime pkgs/main/win-64::vc14_runtime-14.44.35208-h4927774_10 vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.44.35208-ha6b5a95_10 wheel pkgs/main/win-64::wheel-0.45.1-py313haa95532_0 xerces-c pkgs/main/win-64::xerces-c-3.2.4-hd77b12b_1 xz pkgs/main/win-64::xz-5.6.4-h4754444_1 zlib pkgs/main/win-64::zlib-1.3.1-h02ab6af_0 zstd pkgs/main/win-64::zstd-1.5.7-h56299aa_0
xarray	The following packages will be downloaded: package | build ---------- |---------- packaging-25.0 | py313haa95532_1 190 KB pandas-2.3.3 | py313h42c1672_0 13.4 MB xarray-2025.10.1 | py313haa95532_0 2.8 MB ---------- The following NEW packages will be INSTALLED: bottleneck pkgs/main/win-64::bottleneck-1.4.2-py313h2cb717b_0 numexpr pkgs/main/win-64::numexpr-2.11.0-py313h7660c64_1 packaging pkgs/main/win-64::packaging-25.0-py313haa95532_1 pandas pkgs/main/win-64::pandas-2.3.3-py313h42c1672_0 python-dateutil pkgs/main/win-64::python-dateutil-2.9.0post0-py313haa95532_2 python-tzdata pkgs/main/noarch::python-tzdata-2025.2-pyhd3eb1b0_0 pytz pkgs/main/win-64::pytz-2025.2-py313haa95532_0 six pkgs/main/win-64::six-1.17.0-py313haa95532_0 xarray pkgs/main/win-64::xarray-2025.10.1-py313haa95532_0
netCDF4	The following packages will be downloaded: package | build ---------- |---------- netcdf4-1.7.2 | py313h93dd522_2 951 KB ---------- Total: 951 KB The following NEW packages will be INSTALLED: cftime pkgs/main/win-64::cftime-1.6.4-py313h2cb717b_0 netcdf4 pkgs/main/win-64::netcdf4-1.7.2-py313h93dd522_2
matplotlib	The following packages will be downloaded: package | build ---------- |----------

	fonttools-4.60.1 | py313h02ab6af_0 3.6 MB qtbase-6.9.2 | hd965823_2 23.6 MB qtdeclarative-6.9.2 | h88b4c33_1 25.0 MB qtsvg-6.9.2 | h30ace32_1 327 KB qttools-6.9.2 | h7e7b719_1 6.5 MB qtwebchannel-6.9.2 | heb02b0b_1 146 KB qtwebsockets-6.9.2 | heb02b0b_1 134 KB --- Total: 59.3 MB The following NEW packages will be INSTALLED: aom pkgs/main/win-64::aom-3.6.0-hd77b12b_0 contourpy pkgs/main/win-64::contourpy-1.3.1-py313h214f63a_0 cycler pkgs/main/noarch::cycler-0.11.0-pyhd3eb1b0_0 dav1d pkgs/main/win-64::dav1d-1.2.1-h2bbff1b_0 fonttools pkgs/main/win-64::fonttools-4.60.1-py313h02ab6af_0 fribidi pkgs/main/win-64::fribidi-1.0.10-h62dcd97_0 graphite2 pkgs/main/win-64::graphite2-1.3.14-hd77b12b_1 harfbuzz pkgs/main/win-64::harfbuzz-10.2.0-he2f9f60_1 icu pkgs/main/win-64::icu-73.1-h6c2663c_0 kiwisolver pkgs/main/win-64::kiwisolver-1.4.8-py313h5da7b33_0 libavif pkgs/main/win-64::libavif-1.1.1-h827c3e9_0 matplotlib pkgs/main/win-64::matplotlib-3.10.6-py313haa95532_0 matplotlib-base pkgs/main/win-64::matplotlib-base-3.10.6-py313h26e45b9_0 mysql-common pkgs/main/win-64::mysql-common-9.3.0-hf582a5b_3 mysql-libs pkgs/main/win-64::mysql-libs-9.3.0-hc0ebf12_3 pillow pkgs/main/win-64::pillow-11.3.0-py313hb328d1f_0 pyqt pkgs/main/win-64::pyqt-6.9.1-py313h12ec796_0 pyqt6-sip pkgs/main/win-64::pyqt6-sip-13.10.2-py313h630b2a1_0 qtbase pkgs/main/win-64::qtbase-6.9.2-hd965823_2 qtdeclarative pkgs/main/win-64::qtdeclarative-6.9.2-h88b4c33_1 qtsvg pkgs/main/win-64::qtsvg-6.9.2-h30ace32_1 qttools pkgs/main/win-64::qttools-6.9.2-h7e7b719_1 qtwebchannel pkgs/main/win-64::qtwebchannel-6.9.2-heb02b0b_1 qtwebsockets pkgs/main/win-64::qtwebsockets-6.9.2-heb02b0b_1 sip pkgs/main/win-64::sip-6.12.0-py313h706e071_0 tornado pkgs/main/win-64::tornado-6.5.1-py313h827c3e9_0
Celluloid	The following NEW packages will be INSTALLED: celluloid conda-forge/noarch::celluloid-0.2.0-pyhd8ed1ab_0
geopandas	The following packages will be downloaded: package | build ---|--- cffi-2.0.0 | py313h02ab6af_0 293 KB folium-0.20.0 | py313haa95532_0 160 KB scipy-1.16.2 | py313hbd6d557_0 21.7 MB xyzservices-2025.4.0 | py313haa95532_0 65 KB ---

	Total: 22.2 MB The following NEW packages will be INSTALLED: branca pkgs/main/win-64::branca-0.8.1-py313haa95532_0 brotlicffi pkgs/main/win-64::brotlicffi-1.0.9.2-py313h5da7b33_1 cffi pkgs/main/win-64::cffi-2.0.0-py313h02ab6af_0 charset-normalizer pkgs/main/noarch::charset-normalizer-3.3.2-pyhd3eb1b0_0 folium pkgs/main/win-64::folium-0.20.0-py313haa95532_0 geopandas pkgs/main/win-64::geopandas-1.1.1-py313haa95532_0 geopandas-base pkgs/main/win-64::geopandas-base-1.1.1-py313haa95532_0 idna pkgs/main/win-64::idna-3.7-py313haa95532_0 jinja2 pkgs/main/win-64::jinja2-3.1.6-py313haa95532_0 joblib pkgs/main/win-64::joblib-1.5.2-py313haa95532_0 mapclassify pkgs/main/win-64::mapclassify-2.10.0-py313haa95532_0 markupsafe pkgs/main/win-64::markupsafe-3.0.2-py313h827c3e9_0 networkx pkgs/main/win-64::networkx-3.5-py313haa95532_0 pycparser pkgs/main/win-64::pycparser-2.23-py313haa95532_0 pyogrio pkgs/main/win-64::pyogrio-0.10.0-py313he5c4563_0 pyproj pkgs/main/win-64::pyproj-3.6.1-py313h9783e57_1 pysocks pkgs/main/win-64::pysocks-1.7.1-py313haa95532_0 requests pkgs/main/win-64::requests-2.32.5-py313haa95532_0 scikit-learn pkgs/main/win-64::scikit-learn-1.7.2-py313h7f7e138_0 scipy pkgs/main/win-64::scipy-1.16.2-py313hbd6d557_0 shapely pkgs/main/win-64::shapely-2.1.1-py313hbe77627_0 threadpoolctl pkgs/main/win-64::threadpoolctl-3.5.0-py313h4442805_0 urllib3 pkgs/main/win-64::urllib3-2.5.0-py313haa95532_0 win_inet_pton pkgs/main/win-64::win_inet_pton-1.1.0-py313haa95532_0 xyzservices pkgs/main/win-64::xyzservices-2025.4.0-py313haa95532_0

Issues

Issue	***Description***	***Resolution***
Data availability	Ideally, I would have used actual images of sea surface temperature from satellites; however, clouds obscure too many of these images—especially in the region of interest (near the western edge of the meandering Gulf Stream off Cape Hatteras, NC) and desired time frame (winter).	I used hindcasts of modeled data.
Data availability	I found good candidates for animal tracks, but they were from 2015-2016. However, for the environmental data, the latest (and highest-resolution) version of the HYCOM model has hindcasts starting in 2018.	To use the good animal tracks that I found in my region of interest, I had to use an older (lower-resolution) HYCOM model. I also added MUR SST model data, which is high resolution (0.01 degree).
Python environment	My spyder IDE is somehow inconsistent with the python virtual environment I need for the project. I do call spyder from anaconda shell (command line) after activating my virtual environment.	My workaround is: edit in spyder and run in anaconda shell. Note that I have to close plots for the run to proceed…
Animation	When setting the title per frame, I first tried the ax.set_title(title_for_this_frame) for each frame; however, that didn't work; it just gave me a static title. I then tried: initialize the title, save its output to an object e.g. title_text and then use that object's title_text.set_text() method per frame. Title is still static. I tried adding text to the plot—not as title—but this didn't work either…	Skip it
Performance	The MUR SST data was too high resolution (0.01 degree), so it was taking too long to process.	I re-did the download with a "stride" of 4, getting a spatial resolution of 0.04 degrees. This performed much better, and it was still twice the resolution of the HYCOM data.

Future work

Item	***Description***
More recent animal tracks	In the future, I hope to find good candidates for marine mammal tracks that are more recent i.e. withing the time period supported by the higher-resolution HYCOM model (not older than 2018).
Distance to Gulf Stream boundary	For each time sample, define the Gulf Stream boundary (e.g. based upon the spatial temperature gradient). For each point on the track, calculate the distance to the Gulf Stream boundary. Make a time series of these distances over the course of the track.
Try GEE	Instead of downloading all of those data files, try Google Earth Engine, where some of the python code runs locally, but the data stays in the cloud. Since I have some time, I will try this approach.

References

Thorne, L. H., Foley, H. J., Baird, R. W., Webster, D. L., Swaim, Z. T., & Read, A. J. (2017). Movement and foraging behavior of short-finned pilot whales in the Mid-Atlantic Bight: importance of bathymetric features and implications for management. *Marine Ecology. Progress Series (Halstenbek)*, *584*, 245–257. https://doi.org/10.3354/meps12371