HEALPix in the Browser 🚀

How we render HEALPix using JS in the GFTS use case

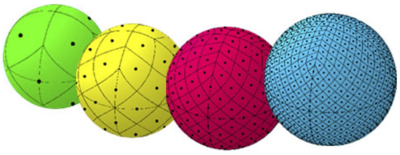
Daniel da Silva[1]
Daniel Wiesmann[1]
Olaf Veerman[1]
Ricardo Mestre[1]
Anne Fouilloux[2]
Tina Odaka[3]
[1] Development Seed
[2] Simula
[3] Ifremer

Introduction

This poster presents the Global Fish Tracking Service (GFTS) use case and our approach to visualizing HEALPix data in web applications. The HEALPix format is widely used in atmospheric modeling, including DestinE Digital Twin data. We demonstrate rendering native HEALPix gridcell data in an interactive Javascript application, providing methodology relevant for the DestinE community.

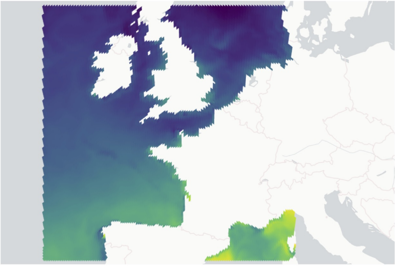
Healpix

HEALPix (Hierarchical Equal Area isoLatitude Pixelisation) is a clever way to divide the entire sky into equal-sized pieces, like cutting a spherical orange into perfectly equal slices. It maps the sphere of the sky onto 12 diamond-shaped sections that can be subdivided into smaller square pixels, with each pixel covering exactly the same area of sky. This makes it perfect for astronomical surveys where you need to analyze data uniformly across the entire celestial sphere.



Alternative methods - XDGGS

There are only very limited options for rendering HEALPix data in the browser. XDGGS is a python package that integrates discrete global grid systems with xarray. It allows for rendering through xarray in jupyter notebooks through lonboard.



The Healpix JS implements healpix functions in javascript, but is not easy to integrate in regular webmap applications.

Our approach

We combined the core healpix functionality provided by Healpix JS package with Deck GL and Mapbox GL to efficiently render healpix data in a webmap application.

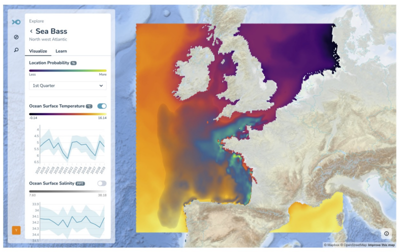
The data to be visualized is stored in parquet files where each row represents a pixel and the columns represent the data values. The HEALPix ID is the first column, and the data variables are stored in the remaining columns. The HEALPix ID is a unique identifier for each pixel in the grid. We then rely on the Parquet JS package to read the parquet files and store the data in a javascript object.

1. Read the data from the parquet files into javascript objects
2. Use the healpix js package to convert the data to lat/lon polygons
3. Use the deckgl package to handle the rendering of the data
4. Use the mapbox gl package to display the data

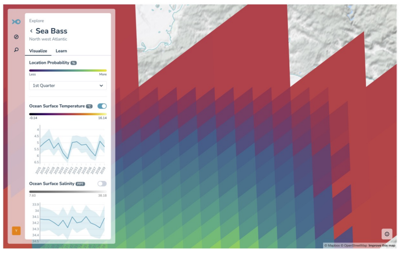
GFTS Case Study

The Global Fish Tracking System (GFTS) allows marine biologists to reconstruct fish tracks from biologging data. Using projections from Destination Earth - eg. sea temperature projections from the Climate Adaptation Digital Twin - we gain a deeper understanding of future fish habitat conditions.

We visualize the fish track data in native healpix format. The screenshot below shows the fish probability density overlaid with the DestinE Climate DT sea surface temperature data.



The healpix are rendered as polygons in the map. The screenshot below shows the overlay, the DT data is two levels above the fish track data.



Next steps

We want to continue to improve healpix rendering in javascript based applications, generalizing the rendering and offering this as an open source package. If you're interested in contributing in any way, please reach out to us.

Resources

https://github.com/michitaro/healpix
https://developmentseed.org/lonboard
https://xdggs.readthedocs.io
https://deck.gl
https://npmjs.com/package/parquetjs