A Study of Air Pollution from the Chernobyl Disaster

Adam Alder

Benjamin Ashton

Spencer Lingwall

Project Repository: https://github.com/USUSciVis/FinalProject.git

Project Overview

The Chernobyl Disaster left a mark of fear on the world from the terrifying effects of the nuclear fallout it caused. Not only was the immediate area destroyed, but areas hundreds of miles away were also affected. By studying the air pollution that came as a result of the disaster we can better inform people of the risks of nuclear power and how large mistakes can affect surrounding areas.

Using this information can help Governments decide what safety measures to put in place for new nuclear power plants. It can also help Engineers and Scientists develop better and safer techniques to harness nuclear power. Understanding the mistakes of the past will help us create a better future for nuclear power.

Additionally, this study will also help us see the flow of particles through the air, giving us more insight into the air currents of the atmosphere and weather patterns. We will be able to better prepare for the effects of events such as these.

Objectives

We aim to visualize how different particles from the fallout travel to the surrounding areas and countries. We also want to be able to see which isotopes travel the furthest and which have the highest concentration of output from the reactor.

Learning how the particles from the reactor travel across large distances and how far the nuclear fallout can affect the area around it can keep people informed about the dangers of nuclear power. Seeing which particles travel the furthest or have the greatest concentration will help people better know what to prepare for in the event of disasters such as this one.

Our primary questions we want answered are:

- Which isotopes have the greatest concentration?
- How far did the particles travel from the nuclear site?
- Is there a wind drift that affected the area? What might the atmospheric currents look like?
- What is a safe distance from a disaster site such as Chernobyl?

Background

Basemap documentation: https://basemaptutorial.readthedocs.io/en/latest/.
VTK File Format: https://kitware.github.io/vtk-examples/site/VTKFileFormats/
Mercadian Map Representation:
https://medium.com/@suverov.dmitriy/how-to-convert-latitude-and-longitude-coordinates-into-pixel-offsets-8461093cb9f5
Radiation Maps for Chernobyl: (Useful for comparing)
https://fromissa-foryou.blogspot.com/2021/05/chernobyl-disaster-map-of-radiation.html

Data and Resources

The data used is from the Chernobyl Disaster Air Concentration dataset from OpenML. This dataset presents concentrations of Iodine-131 (I-131) and Caesium-134 (Cs-134) and Caesium-137 (Cs-137) as aerosol particles which were measured in specific location and date. Location is represented by country, locality, as well as latitude/longitude pairs. In the dataset, time is represented by the date (year/month/day), the time of day that sampling was concluded, and the duration of the sampling.

We used Python as well as multiple Python libraries including both PyVista and OpenVisus to visualize the data.

To generate the gifs we used an online software called Animated GIF Maker:
https://ezgif.com/maker

The dataset can be found at:
https://www.kaggle.com/datasets/debjeetdas/air-concentration-fheaor-the-chernobyl-disaster

Implementation

The first thing we had to do was to clean and preprocess the data. In the beginning we attempted to use VTK data for this and generated the files needed for this, however, this proved rather difficult and we came up with another way by just generating new csv files for the pandas library to read into a Dataframe. Upon doing this we saw that our data had many missing values and some sensors would skip days between data points. To resolve this we linearly interpolated between the days that were given and assumed that any sensor would start at zero if it did not have a start value and would end at zero if it did not have an end value. Additionally some sensor locations had multiple sensors which gave different readings. For these sensors we chose the largest of the two or three values to more accurately portray what was going on in the area.

For the implementation of the 3D render, we used a combination of Basemap and pyvista in a Jupyter notebook. The first thing we do is generate a map of the Europe/Russia area using the python Basemap library. We specifically generated this map using a mercator projection. After this we wrote a bunch of custom functions that would take in a longitude and latitude and convert it into x and y pixel locations given that our map was a mercator projection. We then plotted the mercator map of the area in pyvista and added two sliders. One slider controls the day, and the other controls the isotope data that gets plotted. When the plotter renders the scene, it filters the data by the given isotope and day. Then, for each sensor location, it creates a pyvista box mesh whose height correlates to the concentration of the isotope. This concentration value is also added to the box as a scalar array. The boxes are combined in a MultiBlock and then plotted and colored by the scalar array.

Our second deliverable was a two dimensional interactive map. The interactivity included adjusting the date with a date picker and selecting which isotope to focus on using a dropdown widget. We connected the two widgets to the map by using the ipywidget python library. The color map range was decided by finding the range of values for each isotope. We used the python basemap library to display a detailed map and to plot the values for each sensor. With basemap we were able to pass in longitude and latitude values to easily plot each sensor.

We also made multiple 2d gif animations. These were made by saving images of the 2d interactive map at every date then plugging the images into an online gif generator. 3 types of gifs were made for every isotope. The first was a wind gif showing the spread of the isotopes over time, the second was a heat map of the sensors and their readings of isotope concentrations, and the third was the heatmap, but when the sensor values weren't correlated to color, but opacity.

What did we learn?

Coding Lessons

In this project, we learned how to effectively use two-dimensional datasets in three-dimensional visualizations. We had to create 3D visualizations from scratch, instead of being given 3D datasets and transforming them.

While creating the 3D visualizations we wanted to put the pillars on the correct map location. We originally thought this would be a very simple algorithm where degrees and pixels would convert one to one, however, this was not the case. All maps have distortions and we had to determine how to convert our latitude and longitude coordinates into pixel coordinates. The map projection that we chose to learn how to do this on was Mercator. It is a ‘cylindrical’ map projection of a sphere that stretches to infinity at the poles. While converting the longitude to an x coordinate for our pixels was relatively simple, converting the latitude into its corresponding y coordinate was rather difficult. We first had to create an imaginary larger world map and then calculate the pixel coordinates on that which then allowed us to figure out where on our smaller map the pillar would need to be placed. We tried a few other map projections and Mercator seemed to be the most accurate for our purposes.

We also learned about how vtk files work underneath the hood. Our original implementation plan called for a time series of vtk files, and so we learned how to turn the dataset into vtk format, even though we scrapped this approach later on.

Radiation Lessons

Looking at the visualizations of our data, we learned that there is a significant risk for valleys and mountain bound regions as the isotopes do not have a clear way to escape and spread out more. The highest concentrations were in the region of the Alps where the isotopes sat for several days while other areas only had small spikes that lasted for only a few at most. Our conclusion for where a safe area for a nuclear fallout is in open areas where the air can spread the radioactive isotopes out creating a much less dangerous concentration. In areas where the air can get trapped, there is a much greater risk from the radioactive isotopes.

In answer to our question about if wind drift affected the isotopes, it did not appear to have any effect. All areas seemed to be affected by the isotopes, but like mentioned above, areas where the isotopes could get trapped ended up with higher concentrations of them. There didn’t appear to be a major direction that the isotopes were traveling, however, we also did not have data from the Russia area to completely determine this.

For a ‘safe’ distance from chernobyl we determined that there really wasn’t one if you were trying to avoid all radioactive isotopes. However, by around 1500 miles, the concentrations appeared to be small. This effectively puts most of Europe in a danger zone. Even then those areas that are outside this range were still receiving radioactive fallout from the explosion. As mentioned above, the safest place is actually to be in a large open area where the air doesn’t get trapped in one place increasing the concentration of the isotopes you would be affected by.

For an application of our results, building a nuclear power plant in the rocky mountain region of the United States would be a rather risky idea. This is because the radioactive isotopes from a fallout would get trapped in the valleys and cause higher concentrations than if it were built in the midwest areas. This is not to say that a nuclear power plant in the rocky mountains is a bad idea, but that it carries a greater risk with it due to the mountainous terrain. Engineers should take extra precautions while designing reactors for this area to reduce the risk of a nuclear fallout to as little as possible.

Changes

We had planned on having more two dimensional charts (e.g. a timeline) to analyze the two dimensional characteristics of the data. Instead we incorporated this data into the map visualizations. We did this by using colors, arrows, size, interactive widgets, and animations to show magnitude, direction, and change over time.

We created animations to show change over time and three dimensional visualizations for a different perspective of the data. Neither of these visualizations were in our original plan.

Originally, when we started this project, we had planned on turning the dataset into vtk files for use with pyvista and paraview. We spent the first week learning about and creating vtk files and a time series from the vtk files. However, we revised our plan and instead used the data directly when creating the widgets and renders. That is why vtk file related topics are mentioned in the report but not present in the final code.

Project Evaluation

When writing our proposal, we identified the following four requirements as the measure we are going to use when determining the success of our project.

The update time for the visualizations are reasonable allowing interaction without blocking due to render complexity.

For the 3D render, we met this criteria. The render is fast and smooth, even on laptops, and changing the day or isotope updates quickly. The interactive 2D visualizations do take a few seconds to update, however, this should be within reasonable wait time standards.

The project has interactability to let the user play around and explore the dataset by means of the visualizations.

This is true. We created multiple visualizations that utilize interactive elements to allow the user to explore the data. These interactive elements include widgets allowing the user to change the date, isotope type, and color map range.

The visualizations are accurate to the data.

Short of any human error, the visualizations accurately represent the data. We cross referenced the output of some of our visualizations with other visualizations found online, and were able to verify accuracy.

The visualizations are appealing and are intuitive to understand.

We worked to keep the visualizations simple without too much to distract from the information presented. For the interactive visualizations we used widgets that are easy to use and understand.

The strength of our project is in our variety of visualizations and the weakness would be the time it takes to run the code. We believe that we have met all of the requirements we set for ourselves.

Distribution of Work

Adam Alder preprocessed the data to make it easier to work with in our visualizations and added color mapping to the 3D render as well as optimizing parts of the code to make it run faster. Spencer Lingwall created the initial visualization of Chernobyl and sensor points on a map as well as handling all mercator projection related functions. Benjamin Ashton then created an interactive visualization that allowed the user to see sensor data for individual days as well as shift the color scale to highlight the desired values.

Note: These are just the main points of the project. Each of the group members had an impact on other parts of the project.