

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Weather Video

João Santos Gama Caldas

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Magalhães da Silva Pinto de Andrade

October 3, 2022

Resumo

A meteorologia e a climatologia envolvem o manuseamento de grandes quantidades de informação, quer de histórico ou mesmo de previsão. Por dia são geradas informações provenientes de muitas fontes desde, nomeadamente, satélite às Estações Meteorológicas Automáticas (EMA). Estas gravam várias variáveis meteorológicas desde a temperatura, humidade, intensidade e direcção do vento, precipitação e tipo de precipitação. A gestão de todo este volume e variedade de dados requer ferramentas que tornem este processo de análise e manipulação simples, rápido e eficiente. Atualmente os vários sistemas de representação meteorológica usam Sistemas de Informação Geográficos (SIG) com camadas de informação guardados no formato bitmap comprimidas sendo possível navegar na dimensão temporal. O armazenamento da informação sobre a forma destes formatos é ineficiente do ponto de vista de management de recursos potencialmente prevenindo o utilizador de aceder a todas as variáveis mencionadas afetando assim a sua experiência. O trabalho desenvolvido nesta dissertação pretende encontrar uma solução para as limitações mencionadas. Para tal foi definido o objectivo de conceber um sistema de gestão e visualização de informação meteorológica baseado num formato vídeo em alternativa aos formatos utilizados actualmente. O resultado é o protótipo de um SIG que utiliza vídeos para visualização dos relâmpagos capturados no dia 6 de Abril 2022 na Europa e uma redução de cerca de 40% do espaço necessário para armazenar os ficheiros.

Keywords: Compressão de imagem e vídeo, Streaming de imagem e vídeo, Sistema de informação geográficos.

Abstract

Meteorology and climatology involve handling large amounts of information, either historical or forecast. Each day, information is generated from many sources, mainly from satellites to Automatic Weather Stations (AWS). These record several meteorological variables from temperature, humidity, wind intensity and direction, precipitation and type of precipitation. In order to manage all this information, tools are needed to make this process of analysis and manipulation simple, fast and efficient. Currently, the various weather representation systems use Geographical Information Systems (GIS) with layers of information stored in compressed bitmaps with the option to navigate in the temporal dimension. The storage of the information within these formats is inefficient resource management wise and potentially prevents the user from accessing every variable mentioned affecting their overall experience. The work developed in this thesis seeks to find a solution for the mentioned setbacks of the current systems. With this in mind, a goal was set to develop a management and visualisation of weather information system based on video formats instead of the currently used formats. The result is the prototype of a GIS that displays video for visualisation of all lightning occurrences on the 6th of April 2022 in Europe and a reduction of 40% of the space needed to store the files.

Keywords: Image and video compression, Image and video streaming, Geographical information system.

Acknowledgements

To my supervisor Professor Maria Teresa Andrade, for her experience and guidance. Her support was most useful and imperative to the final result of this thesis's project and writing.

To Engineer Vitor Teixeira and Engineer Diogo Pereira, both from iClimate Adviser, who guided me through the uncharted waters of this dissertation's development and writing. They endured my inexperience during all of the weekly meetings and for giving me the best possible council when facing the issues I ran into while working on this project.

To my parents, family and friends, whose determination always inspired me to push myself and their support during a stressful part of my life.

I cannot possibly convey in any shorter way how thankful I am to all the people who helped me when I most needed it.

João Caldas

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives and Expected Results	2
1.3	Structure	2
2	State of the Art	3
2.1	Introduction	3
2.2	Image Formats	3
2.2.1	JPEG	4
2.2.2	JPEG2000	4
2.2.3	GIF	4
2.2.4	PNG	4
2.2.5	WebP	5
2.3	Video Compression	5
2.3.1	H.264	5
2.3.2	H.265	6
2.3.3	VP9	6
2.3.4	AV1	6
2.4	Computer Vision Libraries	6
2.4.1	OpenCV	6
2.4.2	Matlab	7
2.5	Geographical information systems	7
2.5.1	Earth's Nullschool	7
2.5.2	Windy	7
2.5.3	ZoomEarth	8
2.5.4	IPMA	8
2.6	Web Mapping Libraries	8
2.6.1	Leaflet	9
2.6.2	Open Layers	9
2.7	Map Projections	9
2.7.1	Mercator Projection	10
2.7.2	Web Mercator Projection	10
2.7.3	Albers Projection	10
2.7.4	Azimuthal Equidistant Projection	11
2.7.5	Dymaxion Projection	11
2.8	Chapter Summary	12

3	Development	15
3.1	Introduction	15
3.2	Video compilation	16
3.2.1	Collecting data	16
3.2.2	Compiling to H.264	16
3.2.3	Transparency	17
3.3	Geographical information systems	18
3.3.1	Video hosting	19
3.3.2	Video Overlay	20
3.3.3	Handling Events	20
3.4	Chapter Summary and Use Cases	21
4	Results and Future	27
4.1	Results	27
4.2	Future developments	28
	References	29

List of Figures

2.1	Earth's Nullschool UI	8
2.2	Windy UI	9
2.3	ZoomEarth UI	10
2.4	IPMA UI	11
2.5	Mercator Projection	12
2.6	Albers Projection	13
2.7	Azimuthal Equidistant Projection	14
2.8	Dymaxion Projection	14
3.1	Block diagram of the development process	15
3.2	Block diagram of the development process	17
3.3	Original frame	18
3.4	Mask used	19
3.5	Frame after applying the mask, bodies of water removed	20
3.6	Frame after applying the mask, continent and islands removed	21
3.7	Imgur hosting the different videos	23
3.8	Manually assigned sets of coordinates to videos	23
3.9	Manually assigned links to videos	24
3.10	Result of the application developed - Satellite map mode	24
3.11	Result of the application developed - GoogleMaps map mode	25
3.12	Current Use Case UML diagram	25
4.1	Fully automated Use Case UML diagram	28

List of Tables

1.1	Obtaining the pixel size of image	1
1.2	Obtaining the JPEG file size	1
1.3	Total file size for an entire day and week	2

Abbreviations

2-D	Two Dimensional
3-D	Three Dimensional
AMS	Automatic Weather Stations
API	Application Programming Interface
AVC	Advanced Video Coding
GIF	Graphics Interchange Format
GIS	Geographical Information System
GUI	Graphical User Interface
HEVC	High Efficiency Video Coding
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
PNG	Portable Network Graphics
UI	User Interface
UN	United Nations
URL	Uniform Resource Locator
WWW	<i>World Wide Web</i>
WebP	Web Picture format

Chapter 1

Introduction

1.1 Motivation

Images and video reach substantial sizes leading to a need for more storage and more significant transfer bandwidths. Using this thesis's focus as an example: the standard resolution for these systems is 1 pixel every 0.5° on a Mercator Projection map (a $180^\circ \times 360^\circ$ rectangle). This single pixel is equivalent to a 3000km^2 area. That results in an image with a total of 76 kilobytes using JPEG compression, keeping in mind that JPEG is an RGB-based format, so each of the three colour channels have 8 bits assigned. Assuming a rate of 1 image every hour for an entire week, we would have 12.5 megabytes worth of frames for a single weather variable map. Every value in this example was calculated using the following method.

		Resolution		
		1 pixel per x degree		
Map height (degrees)	180	0.5	360	Map height (pixels)
Map width (degrees)	360	0.5	720	Map height (pixels)
			259200	Total pixels

Table 1.1: Obtaining the pixel size of image

To determine the number of pixels needed for this image, it is necessary to divide the height and width of the figure by the resolution. The total size is the result of the product of these values.

RGB channels	3		259200	Total pixels
Bits per channel	8			
JPEG Compression	0.1		75.9375	Total size (kB)

Table 1.2: Obtaining the JPEG file size

JPEG format is RGB. The three channels use 8 bits to store the information for a total of 24 bits or 3 Bytes after converting. After converting the pixel size of the image to Bytes (dividing by

2^{10}), the product of this value by the colour channels and the JPEG compression (around 10% of the size) we obtain the value of a single image.

	Total images		75.9375	Total size (kB)
1 day	24		1822.5	Total size (kB)
1 week	168		12.45849609	Total size (mB)

Table 1.3: Total file size for an entire day and week

Assuming one image per hour, the estimated file size is the product of the single image file size and the number of hours needed. In reality, it is expected to be more than this amount in a day to do a proper study,

These values can be further compressed. After analysing some of these maps, it is possible to recognise the similarity between each frame. Repeatedly storing these similarities can prove to be a waste. If we compile these frames into a video, we can hypothetically lower its size due to image interpolation, a prevalent method in video compiling. This method alone makes it less of a resource burden, especially when streaming.

The nature of this project is heavily exploratory, as so it is expected large amounts of manual experiments and trial and error, before achieving a state of the development stage where it is possible to automate these tasks.

1.2 Objectives and Expected Results

The main goal of this project was to create an application for a Geographical Information System displaying the information in video format instead of the image-based current state of the art. Ideally, this new system would prove to be less resource intensive to the back-end while being capable of providing the same information as any existing GIS. It should be more efficient and user-friendly. Results are discussed in their own chapter.

1.3 Structure

The report has been divided into four major sections. Chapter two focuses on the current state of the art. The development chapter explains the different stages of the project, the difficulties faced, and the solutions found. Results and Future provides a conclusion to this report and a reflection on potential features that could improve the final product.

Chapter 2

State of the Art

2.1 Introduction

This chapter of the report focuses on the current state of the art. It is divided into six subsections. The first two discuss different types of images and video compression by explaining how different formats handle the information. The Graphical User Interface section explains the most up-to-date Geographical information systems and how they present their data. These systems offer a large amount of information and will heavily determine the prototype's and final product's presentation and features. The following sections present a small overview of some libraries that will be used to develop the project, as well as an explanation of how the Earth is projected on maps, virtually or not. Lastly, a summary of the tools used during the project's development.

2.2 Image Formats

Image formats store the data of a digital still image. This thesis will focus on Raster formats, that is, formats that represent images employing a grid of pixels, unlike vector formats that create shapes with the use of, for example, points, lines, and curves to represent the image on a Cartesian plane [1]. Image Formats are defined by different properties such as image compression to allow for more efficient data storage, transmission, and transparency. Each format uses different but sometimes similar algorithms that result in either lossy or lossless compression. Lossless compression is used when it is imperative to save every detail associated with the image, i.e., images that can be perfectly reconstructed, unlike lossy compression, where only an approximation of the original can be guaranteed, where the difference should be insignificant [2]. On the other hand, examples of lossless algorithms include entropy encoding (Shannon and Huffman coding) and adaptive dictionaries (like Lempel–Ziv–Welch). Lossy algorithms include but are not limited to transform coding, chroma subsampling and fractal compression. Lossless compression is typically used for video or audio production, while lossy is used when the file size is relevant to the overall user experience [2]. Formats that handle transparency normally include an extra bit with relevant

data. There are several formats available, each with their advantages and selecting a particular format over the others is critical to this thesis.

2.2.1 JPEG

JPEG stands for "Joint Photographic Experts Group". This format was developed as a collaborative effort of different companies and groups starting in 1986 and was released in September 1992. It is a widely used format across the Internet and is one of the most popular image formats for digital images and photos. It uses a lossy compression algorithm, but the user can adjust the degree of compression. It uses the Discrete Cosine Transform (similar to the Discrete Fourier Transform), where the data is represented as a sum of cosine waves of different frequencies that are then further compressed using Huffman coding for its compression algorithm. Each RGB channel is stored separately. According to NASA, compression of 1:120 still holds a high value of acceptability [3].

2.2.2 JPEG2000

This format was developed by the same committee that developed the original JPEG between 1997 and 2000. It was created with the objective of having a more versatile format with similar compression quality and performance. There are multiple ways to decode this format, making it flexible and efficient in numerous circumstances. Because of this flexibility, it requires more and more complex codecs that are resource-demanding. Its compression is based on the Discrete Wavelet Transform (DWT) algorithm. Despite being selected as the coding standard for digital cinema [4], it is not supported by most web browsers and is not widely used on the Internet.

2.2.3 GIF

Graphics Interchange Format is a popular format commonly used for simple animations nowadays. It was first released in June 1987. This format uses the Lempel–Ziv–Welch (LZW) lossless data algorithm [5], the same algorithm used for file compression in UNIX systems. It is a simple method that works by using coding dictionaries. It scans the data and creates a model of its statistical structure in real-time. This technique reduces the file size, maintaining the image's visual quality.

2.2.4 PNG

In 1994 GIF's compression algorithm (LZW) was patented, so there was a need to develop a free-to-use lossless compression image format [6]. It uses the Deflate algorithm, the same used for file compression for ZIP files, a combination of Huffman coding and LZ77, the algorithm where LZW originates from. This format is unique from the other mentioned image format because, apart from the three colour layers (red, green and blue), it also features an alpha layer that provides

transparency for the image for both grayscale and full RGB images. PNG offers two different RGB palettes (24-bit RGB and 32-bit RGBA) yet does not support other colour models like CMYK.

2.2.5 WebP

WebP was developed by Google in 2010 as a substitute format for the most popular and universally used JPEG, GIF and PNG [7]. Its most recent version supports both lossy and lossless compression, animations and alpha layer, condensing the most relevant features of the mentioned formats into one while providing better image quality for the same size. The most prominent browsers and all Google products have WebP support. Its lossy compression method uses intra-frame VP8 key frame encoding to compress image data, a commonly used video compression technique. Pixels are commonly very similar to their neighbouring pixels so they are grouped inside the same code block and stored using fewer bits. This way proves to be a more efficient system resource-wise. On the other hand, its lossless algorithm is based on entropy coding for each different colour channel, dictionary coding and Huffman coding.

2.3 Video Compression

Video formats store digital video data typically as a sequence of images with a temporal offset (40ms for a 25fps frame rate). These formats use codecs and algorithms that encode and decode information like light and sound to compress the file size. Most codecs work by block-matching. They go through every frame (every picture that makes the video) and match what they have in common, thus reducing its total size by not wasting data in repeated information either inter frame, due to sampling on the temporal sequence, or intra frame, spatial sampling. Factors like resolution determine the number of samples needed. High-Definition Television broadcast (around 1440x1152) requires more than 1.3 million sampling points [8]. For a captured scene to look natural to the human eye, it has to be recorded with at least 20 frames per second frame rate. The higher the frame rate, the smoother the movement is, although it requires more file size.

2.3.1 H.264

H.264, also known as Advanced Video Coding (AVC) or MPEG-4, is a prevalent format, achieving 92% of the recording, compression, and distribution of digital video in 2018, according to Bitmovin¹ assuring this way its status as an industry standard. It was developed by the Video Coding Experts Group with the objective of creating a more efficient (bitrate-wise) but still high-quality video format compared to its 1998 contemporaries. A first working version was drafted in 1999. H.264 works similar to some raster files, processing a grid of pixels to create subsections of each frame called macroblocks. They store coded data for a 16x16 region for each frame [9].

¹<https://go.bitmovin.com/hubfs/Bitmovin-Video-Developer-Report-2018.pdf>

2.3.2 H.265

H.265, also known as High Efficiency Video Coding(HEVC), is a more advanced version of the H.264 format. It uses a more advanced algorithm so that a better quality resolution can be achieved easily by comparing both formats for a comparatively smaller bandwidth. Unlike AVC, which uses macroblocks, HEVC processes the frames with coding tree units that use bigger block structure samples that can be dynamically sub-partitioned [10]. The dynamic nature of this process achieves better coding efficiency.

2.3.3 VP9

VP9 is an open-source video coding format designed for more efficient video streaming over the Internet. It was developed to obtain better compression efficiency than HEVC, and unlike the former, it is currently supported by most modern web browsers. The VP9 encoder has a two-pass rate-control encoding option which proves especially useful for low bit-rate video streaming [11]. It also resorts to intra-frame coding for its compiling. Since neighbouring pixels within an image are often very similar, the frame image is divided into blocks rather than storing each pixel independently. The typically minor difference between each pixel can be encoded using fewer bits.

2.3.4 AV1

AOMedia Video 1, commonly referred to as AV1, is an open, royalty-free video coding format conceived as a successor to VP9 in 2018. Similar to other image and video formats, it uses macroblocks, entropy encoding, and intra-frame coding, among other methods, to compile a video. Compared to VP9, it is 17.11% more efficient in bit-rate savings [12] at the cost of taking significantly more time to encode than any other video format mentioned.

2.4 Computer Vision Libraries

Computers do not interpret images the same way humans do. Computer vision enables computers to process and analyse digital images and videos. Computer Vision Libraries offer an array of useful functions capable of processing and compiling images or video.

2.4.1 OpenCV

OpenCV² (Open Source Computer Vision Library) is a computer vision open-source library mainly used for image, animation and video manipulation. It is also designed for machine learning, accommodating more than 2000 optimised algorithms that can be useful in the future development of this thesis's project. This project's needs demand a tool capable of analysing, masking, editing

²Github repository as of November 28, 2022- <https://github.com/opencv/opencv>

and compiling source frames, obtained in the image format discussed in previous sections, into an output video format.

2.4.2 Matlab

Matlab offers a high-level programming language that easily allows the user to create matrices and arrays especially useful for signal processing, deeplearning and AI. It offers a computer vision toolbox³ that has a vast library of functions and algorithms computer vision-related.

2.5 Geographical information systems

Geographical information systems (GIS) are tools that help users quickly acquire geographic data. They can be used to great effect in both professional and academic [13] settings. Online tools like Earth's NullSchool, Windy and ZoomEarth are perfect examples of working GIS capable of displaying a large amount of geographical and weather information. They represent the current state of the art of this technology and offer various weather-related maps like wind, temperature, and humidity, among others. It is also possible to select the specific date and time of the data. Some GIS users are more likely to want to access local meteorological data. It is then logical that the image files should not be fully loaded for smaller areas. The original image is divided into tiles, and a new image is recompiled client-side with the user's selected area [14]. This method has proven to be successful for other GIS. Similar to the already existing state of the art, this thesis's objective is to adapt this format but using video.

2.5.1 Earth's Nullschool

Earth's Nullschool⁴ displays its information on a globe instead of a map representation of Earth. It is a more realistic visualisation of Earth's climate. Since it is updated every three hours, we can get an accurate representation of the current weather situation at any point on the planet. The user has access to more maps than the standard GIS, making it a powerful tool and most likely a solid basis for the thesis's final project. Earth's Nullschool User Interface is shown in 2.1.

2.5.2 Windy

Windy's⁵ main objective is to provide wind-related data presenting it in the form of animation. Since its release in 2014, it has developed to show current, historical and even predictions for wind patterns. As of 2022, it provides the option to change to other types of weather maps. Windy's User Interface is shown in 2.2.

³Github repository as of November 28, 2022-<https://www.mathworks.com/products/computer-vision.html>

⁴Website as of November 28, 2022- <https://earth.nullschool.net/>

⁵Website as of November 28, 2022- <https://www.windy.com/>

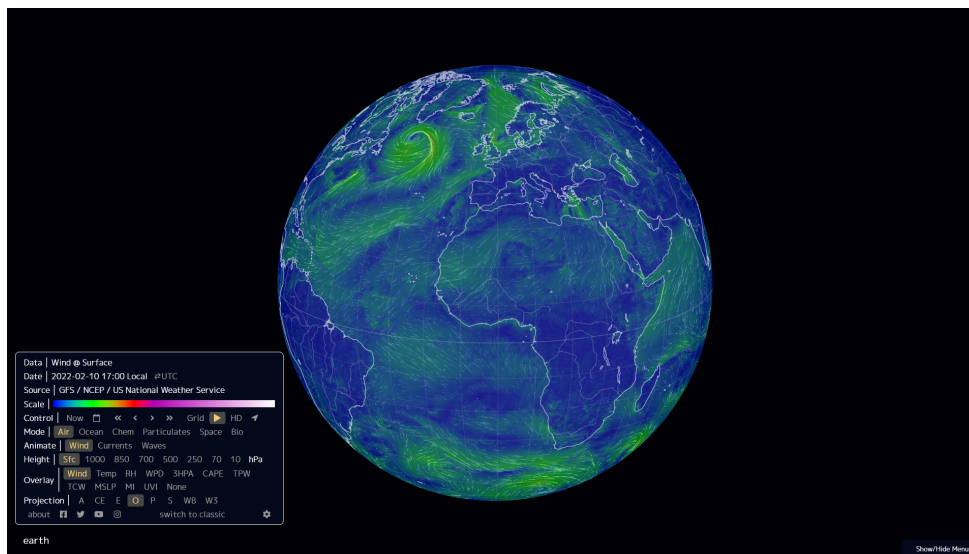


Figure 2.1: Earth's Nullschool UI

2.5.3 ZoomEarth

ZoomEarth⁶ updates every ten minutes based on two geostationary satellites, making it a powerful tool when tracking cyclones and severe storms, wildfires or any type of natural hazard in real-time. ZoomEarth's User Interface is shown in 2.3

2.5.4 IPMA

IPMA⁷ (Instituto Português do Mar e Atmosfera) has a GIS focused on the Portuguese mainland and both Açores and Madeira archipelago. It provides national weather data such as weather forecasts and warns users of potential dangerous climates such as strong winds, fires and tides. IPMA's User Interface is shown in 2.4.

2.6 Web Mapping Libraries

Web mapping displays virtual geographic information allowing the user to interact with the data directly for an overall better user experience [15]. It is critical to analyse the currently available libraries and consider their features, advantages and complexity in order to make a fully informed choice regarding what's the ideal tool to use to fully develop this thesis's GIS prototype.

⁶Website as of November 28, 2022- <https://zoom.earth/>

⁷Website as of November 28, 2022- <https://earth.nullschool.net/>

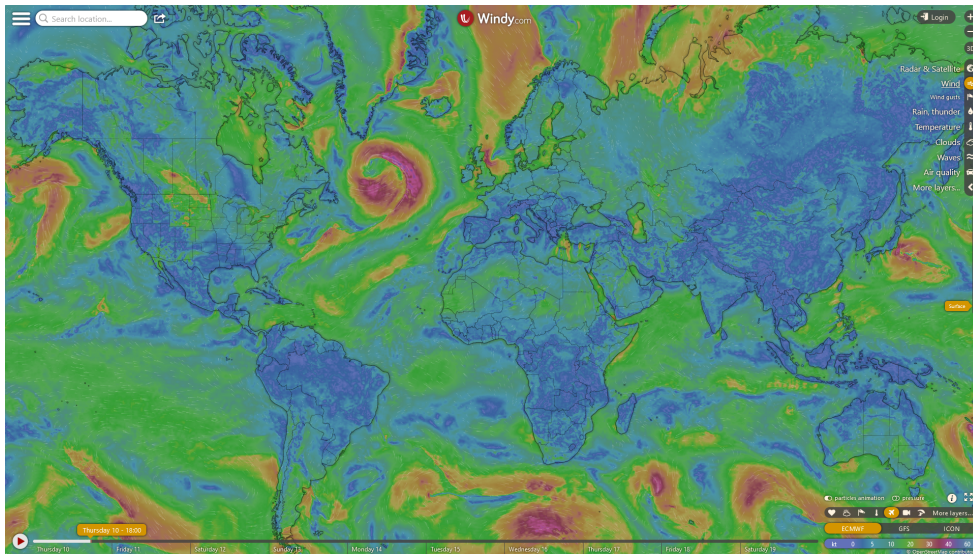


Figure 2.2: Windy UI

2.6.1 Leaflet

Leaflet⁸ is an open-source JavaScript library for web mapping. It has a very detailed and simple-to-use documentation page, so it is perfect for developers without any GIS background to easily project open-source maps into a map object. It comes with HTML and CSS support.

2.6.2 Open Layers

Open Layers⁹ is a more complex open-source JavaScript library⁶. Its larger API provides multiple tools and options to the developer, which only benefits the user. Open Layers is bigger in size than Leaflet; however, its extra features are not deemed useful enough when comparing both libraries.

2.7 Map Projections

3-d spherical shapes cannot be represented on a 2-D plane without sacrificing distance, shape or size, or any combination of the three [16]. The only completely faithful way to represent the Earth is a globe. Despite this fact, 2-D projections are still useful pragmatically speaking. Understanding that no projection is perfect is essential. They are designed to fit different specific situations where the representation of any certain map variable like the ones mentioned is paramount compared to the others. The Mercator projection is historically the most popular projection. Most users should be familiar with it. More specifically, the Web Mercator projection is also the default option for Leaflet, Open Layers, and all the major virtual map applications like Google Maps and OpenStreetMaps use it.

⁸Github repository as of November 28, 2022- <https://github.com/Leaflet/Leaflet>

⁹Github repository as of November 28, 2022- <https://github.com/openlayers/openlayers>



Figure 2.3: ZoomEarth UI

2.7.1 Mercator Projection

This cylindrical-based projection was developed by cartographer Gerardus Mercator in 1569. It mostly preserves the distances and shapes of the areas but distorts their size the closer they are to the poles. This disparity occurs because all parallels are projected to have the same length as the equator when in reality, they become increasingly smaller the closer to 90° N/S latitude. This discrepancy creates the illusion that landmasses near both poles are bigger than what they are in reality. A famous example of this effect is how Africa and Greenland appear to possess similar sizes when in reality, it is 14 times bigger. Its popularity derives from its usefulness in navigation for local charters. Being a conformal map projection means that it preserves relative local angles and shapes [17]. The smaller the scale of the map, the closer it is to representing the area. This is because the smaller the area is, it is easier to ignore Earth's curvature and assume it is a flat 2-D plane. A representation of the Earth following the Mercator Projection can be found in 2.5.¹⁰

2.7.2 Web Mercator Projection

This projection is a special variant of the original Mercator specially designed for virtual map applications. On the global scale, the difference between the two is non-existence for the naked eye, but for local maps, it can deviate from the original [18].

2.7.3 Albers Projection

Contrary to conformal map projections, the Albers projection is an Equal-area projection, meaning that they preserve area, sacrificing shape. This projection is conic-based. It is the second most prominent projection after Mercator [19] because it is preferable to use conic-based projections

¹⁰Daniel R. Strebe. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Mercator_projection_Square.JPG



Figure 2.4: IPMA UI

for extensive distances west to east and regions between the tropics and the poles. This is the case for Europe, Asia and north America, therefore explaining its popularity since these are the historical region for mapmaking. A representation of the Earth following the Albers Projection can be found in 2.6.¹¹

2.7.4 Azimuthal Equidistant Projection

Although not a popular map projection, the Azimuthal Equidistant Projection is recognisable as the map used on the flag of the United Nations following resolution A/107 [20]. This is a very useful projection for determining the distance between two points, one where the map is centred (the north pole in the UN's case) and any other point. A representation of the Earth following the Azimuthal Equidistant Projection can be found in 2.7.¹²

2.7.5 Dymaxion Projection

Also known as Fuller's Projection, this projection has a very different approach compared to the others presented. It projects an icosahedron version of the globe instead of a sphere. This unorthodox method was designed with the objective of fairly distorting all mapping elements such as scale, distance, and shape equally [21]. This projection is recognisable for sometimes being used on maps representing early human migrations. A representation of the Earth following the Dymaxion Projection can be found in 2.8.¹³ .

¹¹Daniel R. Strebe. Wikimedia Commons. https://en.wikipedia.org/wiki/File:Albers_projection_SW.jpg

¹²Daniel R. Strebe. Wikimedia Commons. https://en.wikipedia.org/wiki/File:Azimuthal_equidistant_projection_SW.jpg

¹³Justin Kunimune. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Dymaxion_projection.pdf



Figure 2.5: Mercator Projection

2.8 Chapter Summary

After considering the current state of the art, it is possible to make more informed decisions regarding the development of this thesis's project based on its precedents. The input image-based information should be compiled into video applying either H.264 or H.265 codec. OpenCV library will most likely be used for this process. The output file will be projected on a Web Mercator GIS coded in Javascript with the help of either Leaflet library or Open Layers. A deep understanding of any of these tools is critical for the thesis's success.

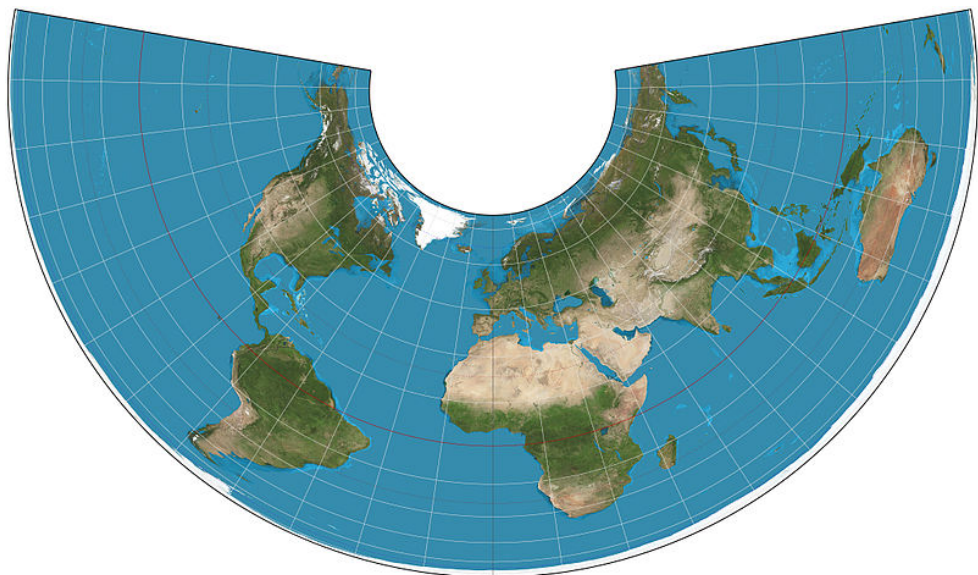


Figure 2.6: Albers Projection

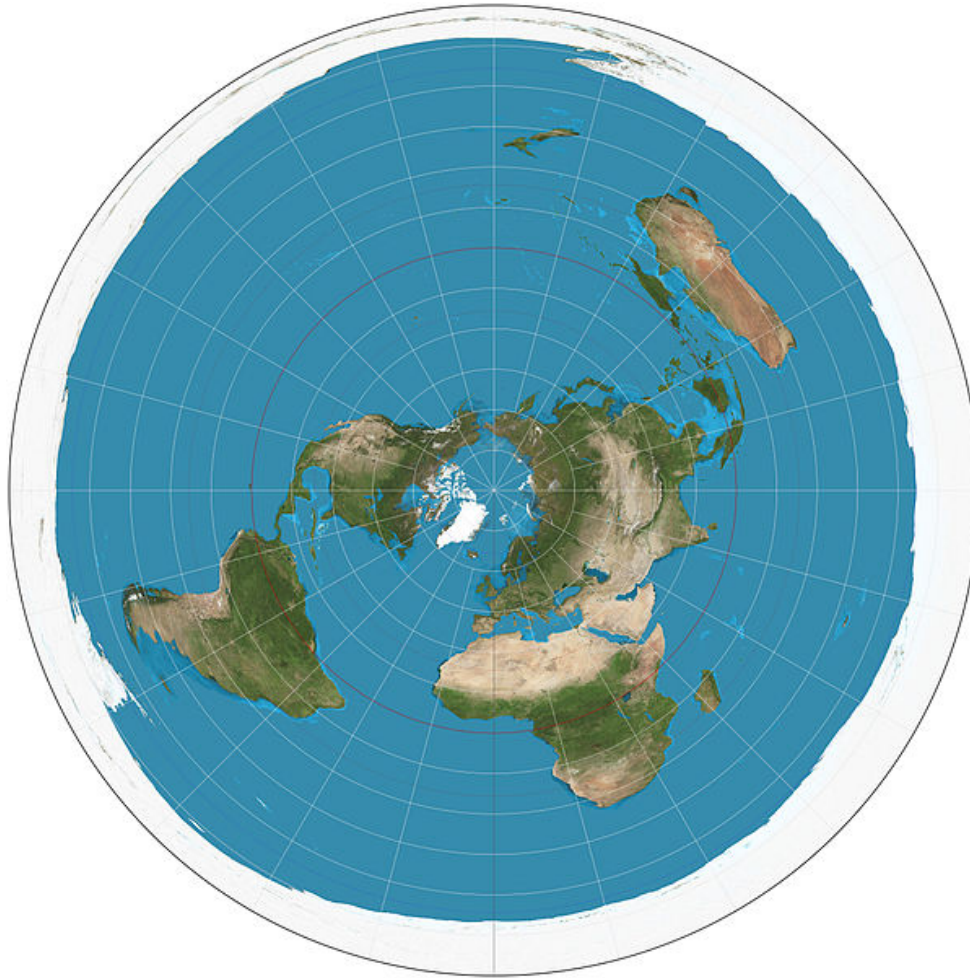


Figure 2.7: Azimuthal Equidistant Projection



Figure 2.8: Dymaxion Projection

Chapter 3

Development

3.1 Introduction

Geographical information systems (GIS) are essential tools for contemporary weather scientists, but they still offer valuable information for any type of user interested in weather forecast. They can typically be accessed by a browser, like the ones mentioned in the state-of-the-art 2.5, or by installing specific software like Google Earth. Similar to these applications, users are expected to be able to perform instinctive interactions like zooming and moving the map, as well as change the map mode or change the time and date of the information displayed. This chapter explains the development process of these features and how the project evolved over the months to the current stage. It analyses the difficulties and the solutions found. It is divided into two major sections: the first one describes the process for video compilation, from the original collection of data, how transparency was handled and how the video is divided into sections. The second section recounts how the thesis's GIS was developed and how the compiled videos were displayed. The last section summarises the chapter. The following block diagram summarises the process for how natural weather phenomena records end up being shown in the prototype. Take lightning and thunderstorms as an example: Blitzortung.org records lightning strikes. This information is readily available and can be downloaded in image form, which is then compiled to be hosted on a video hosting website. They are overlaid on an interactive map and are locally hosted on a browser.

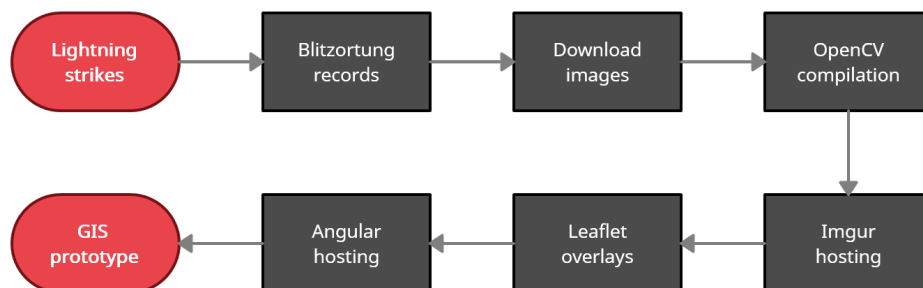


Figure 3.1: Block diagram of the development process

This system was streamlined to be used and managed by two different actors, the first one being the the application user, weather scientist for most cases, even tho the general public being able to access the information and come with their own non-professional opinions. The second actor being the administrator who's current shop requires manually attentions at each of the steps necessary to present the data, from it's manual sourcing, manual video conversion and manual setting of the respective pairs of coordinates of the data on a map.

3.2 Video compilation

This section explains how the videos are compiled using the Python computer vision library OpenCV. OpenCV proved to offer more documentation online, and Python is a computer language I am familiar with.

3.2.1 Collecting data

The original source data was provided by Blitzortung.org. This website tracks lighting and thunderstorms worldwide in real-time and archives them. This data can be accessed and downloaded on either a PNG or GIF format with the height and width, time span per frame, the number of frames, the delay between frames, and the delay of the last frame can be adjusted to what fits the user the best when choosing to export the information as a GIF like shown in [3.2](#).

During the development process, as suggested by engineer Vítor Teixeira, I attended a NASA conference about downloading and sourcing satellite data. These map modes are not used at this stage of the prototype, and their future implementation is discussed in the last chapter of this report [4.2](#).

3.2.2 Compiling to H.264

The 12 source images provided by Blitzortung were downloaded in GIF format. They are 10 seconds long, and each represents an interval of 2 hours of lighting strikes information. Blitzortung.com archive data does not allow the download of more extensive intervals of data. The data displayed in the prototype was tracked on the 6 of April 2022. The Python program only considers the last six source files (from 12h to 24h of that day). Together they result in a final output with exactly 1 minute of length. These GIFs are divided into their respective frames, and each frame is divided then into 64 equally sized square-shaped videos. The videos are compiled using the H.264 codec. This proved to be the only option for compiling the video. The reasons why as well as for the specific length are explained in [3.3.1](#). As mentioned in the state-of-the-art chapter, GIS systems only display what the user sees at any moment in order to waste resources as less resources as possible. In order to achieve this effect, the map is divided into several square tiles that are only activated when the user can see them. For the purposes of better presentation, the examples shown only show four divisions instead of the 64. The shape of these divisions is based on how the current state of art GIS that also uses a grid system to display the information. Other

Selection: *

End time: Date: Time (UTC):

Start time: Date: Time (UTC):

Raw data (only for the contributors)
Geographic restriction:

North: °
West: ° East: °
South: °

Image
Map selection:

Width adjustment of raw map: px
Width adjustment of resulting map: px

Without legend:
Without statistic:

png-Image
 gif-Image (animated, only available on days with lower server load)

Time span per frame: min
Number of Frames:
Delay between frames: ms
Delay last frame: ms

Figure 3.2: Block diagram of the development process

alternatives were tested like the division in multiples rows but the continental shape and borders did not line up.

3.2.3 Transparency

Applying transparency to the video was a feature tackled early on. The plan was to apply a mask to remove the sea background that the source files have before the frame is divided into the sections, as doing it 64 times would lead to longer time spent on the compilation. This proved to work for individual frames.

The mask is automatically generated in the program. It analyses all the colours found in each frame, and after determining the most common one, it creates a selection of all pixels with colour

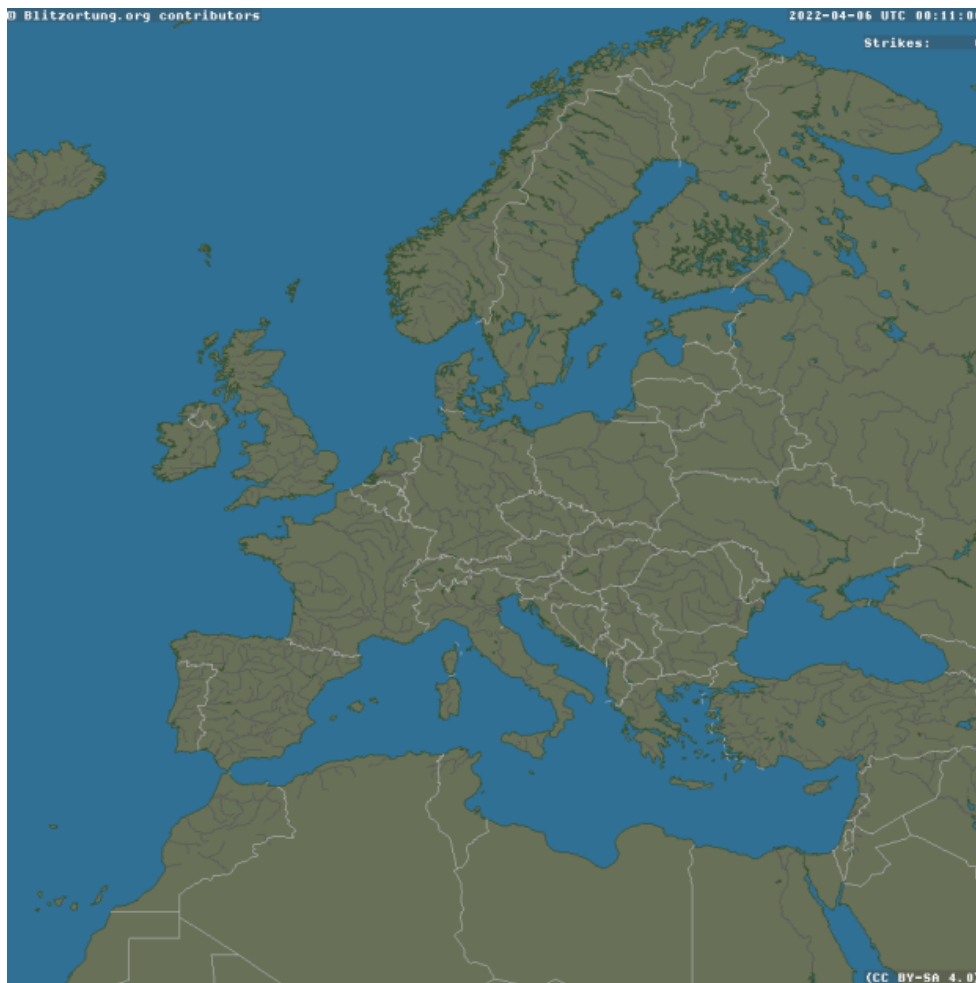


Figure 3.3: Original frame

and similar colours with a small margin. As we can observe from the example provided in 4.1, the background of the image is a tone of blue that represents the ocean, seas, rivers and lakes around and within Europe. 3.4 is the result of the mask generation.

The mask can be applied in two ways to the original image. It is possible to use it to keep only the landmass, as seen in 3.5 or, inversely, only keep the bodies of water 3.12.

Ultimately the transparency feature is not used on the prototype since the H.264 format does not support alpha-layer.

3.3 Geographical information systems

This section explains how the videos compiled in the previous section are displayed. Considering the fact that I had already used the Javascript library Leaflet for a similar project last semester, I



Figure 3.4: Mask used

decided to use my experience with this tool for this project. Leaflet was designed with simplicity in mind according to their own website¹ and provides tutorials, an API reference and plugins so that amateurs GIS designers can have a simpler time building their application. Leaflet's maps are fully interactable. The user can change map modes, and actions like changing the zoom or moving around the map are tracked through events that can be fully customisable. The application is locally hosted with the help of AngularJS. This tool is a framework developed by Google for simple Javascript, HTML and CSS website design.

3.3.1 Video hosting

In order to overlay images or videos on Leaflet requires an URL of the file. The videos were hosted on imgur.com, where all videos uploaded are automatically converted to H.264 and the maximum length of a video uploaded is 1 minute. The final 64 videos can be accessed at <https://imgur.com/a/G127uOx>

¹As of November 28, 2022- <https://leafletjs.com/>



Figure 3.5: Frame after applying the mask, bodies of water removed

3.3.2 Video Overlay

Leaflet requires two sets of coordinate^{3.8} and an external URL^{3.9} of the source in order to import and display any media format. Both have to be inputted manually. As mentioned in [3.3.1](#), the videos are hosted on imgur.com. The two sets of coordinates define the top left corner and bottom right corner of the real-world area where the video should be overlaid on top.

3.3.3 Handling Events

Leaflet offers a method to track the user's interactions with the map by moving it or zooming in. Every time the user performs one of these actions, Leaflet makes sure only to display the videos loaded on-screen by intersecting the area seen with the area provided when loading the videos in [3.3.2](#). This way, it works similarly to the other GIS in the state-of-the-art chapter, where the objective is to economise the amount of resources used since these applications handle a substantial quantity of information.



Figure 3.6: Frame after applying the mask, continent and islands removed

3.4 Chapter Summary and Use Cases

The result of the development process cannot be faithfully displayed since the images only capture a moment of the video footage. The following two figures represent the result of the prototype developed for this thesis.

Use Case:	Interactive map
Actor:	User(all)
Purpose:	A more innate User Interface
Overview:	By dragging the map it moves. By scrolling it zooms. Selecting another background changes it. These are common features for most Geographical Information Systems and other online map related applications and are implemented with Leaflet
Priority:	Primary

Use Case:	Change climate mapmode
Actor:	User(all)
Purpose:	Compare information form different mapmodes in order to find patterns and determine professional conclusions based on data or for casual used
Overview:	The user should be able to easily change from seeing a temperature mapmode to a humidity mapmode. The backend must be able to provide both and the frontend an easy interface
Priority:	Primary

Use Case:	Review historical Data
Actor:	User(weather scientists)
Purpose:	Analyse long and short term climate activity. The more information the user is provided, the higher the chances of predicting the weather or
Overview:	The user should be able to easily change when the information was captured, normally with the use of at timeline. The backend must be able to provide both and the frontend an easy interface
Priority:	Primary

Use Case:	Manually maintaining the backend
Actor:	User(administrator)
Purpose:	Give the user more options
Overview:	At the current state of the prototype, an human actor has too manually manage the information available to the user actor by sourcing, compiling and displaying all available data.
Priority:	Primary

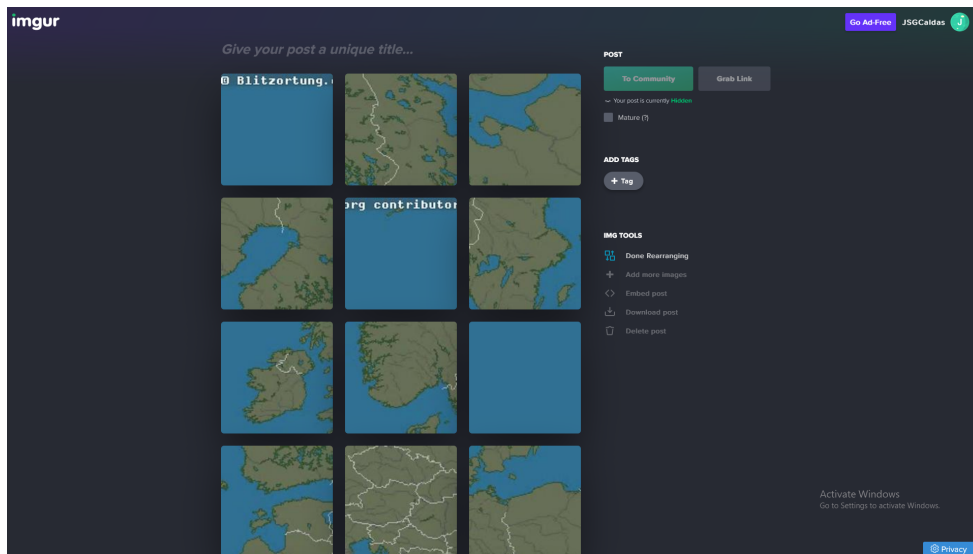


Figure 3.7: Imgur hosting the different videos

```

var rec = [
  // L.latLngBounds([71, -20], [68, 44]),
  // L.latLngBounds([71, -30], [60, 50]),
  // L.latLngBounds([51, -20], [22, 12]),
  // L.latLngBounds([51, 12], [22, 44]),
  // L.latLngBounds([51, 12], [22, 44]),
  // L.latLngBounds([51, 12], [22, 44]),
  //L.latLngBounds([60, -50], [20, 50]),
  //L.latLngBounds([31.5, -20], [20, 44])
  L.latLngBounds([72, -20], [43, 12]),
  L.latLngBounds([72, 12], [43, 44]),
  L.latLngBounds([51, -20], [22, 12]),
  L.latLngBounds([51, 12], [22, 44])
]

```

Figure 3.8: Manually assigned sets of coordinates to videos

```
var videoUrls = [  
  // 'https://www.mapbox.com/bites/00188/patricia_nasa.webm',  
  // 'https://www.mapbox.com/bites/00188/patricia_nasa.mp4'  
  // 'https://i.imgur.com/OPRnFDj.mp4'  
  // 'https://i.imgur.com/Ait7vpY.mp4'  
  'https://i.imgur.com/eRy5x4o.mp4', //NW Europe  
  'https://i.imgur.com/LtfFyQo.mp4', //NE Europe  
  'https://i.imgur.com/C7tfRqV.mp4', //SW Europe  
  'https://i.imgur.com/0n3NWJC.mp4', //SE Europe  
  // 'file:///home/jcaldas/Documents/Tese/weathervideo/opencv/output/europecut0.mp4',  
  // 'file:///home/jcaldas/Documents/Tese/weathervideo/opencv/output/europecut1.mp4',  
  // 'file:///home/jcaldas/Documents/Tese/weathervideo/opencv/output/europecut2.mp4',  
  // 'file:///home/jcaldas/Documents/Tese/weathervideo/opencv/output/europecut3.mp4'  
  // 'https://i.imgur.com/KDa0EMf.mp4',  
  // 'https://i.imgur.com/Ze8Zf00.mp4',  
  // 'https://i.imgur.com/Mw6U7vI.mp4',  
  // 'https://i.imgur.com/zureEea.mp4',  
  // 'https://i.imgur.com/ahk55Q0.mp4',  
  // 'https://i.imgur.com/NFD18HM.mp4',  
  // 'https://i.imgur.com/KNx7Lh4.mp4',  
  // 'https://i.imgur.com/bNwYJSP.mp4',  
  // https://imgur.com/a/10euTIL  
  // https://imgur.com/a/ghDTLU7  
];
```

Figure 3.9: Manually assigned links to videos

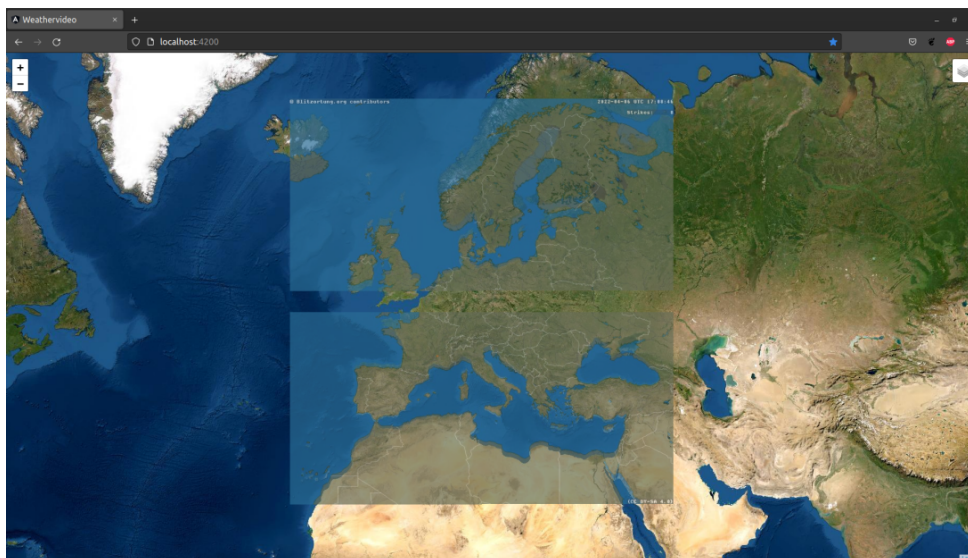


Figure 3.10: Result of the application developed - Satellite map mode

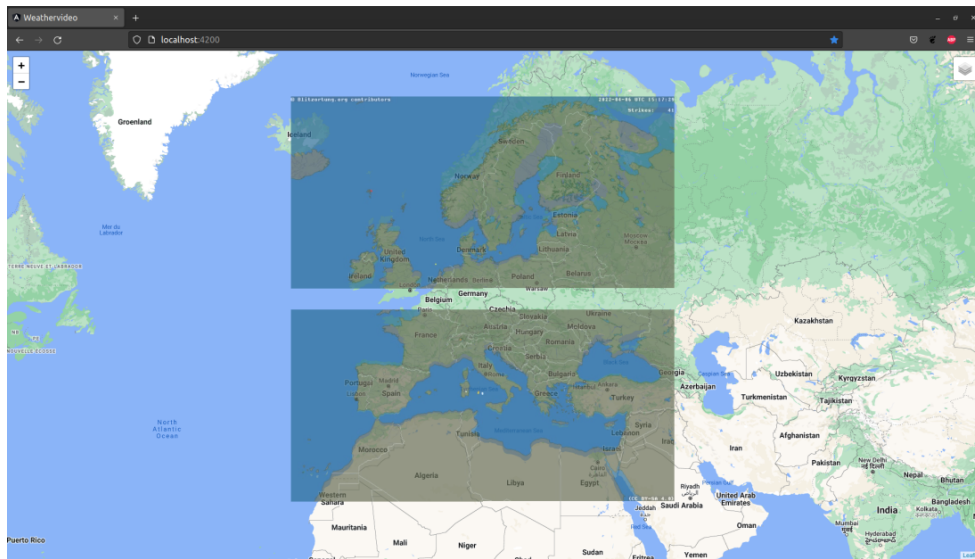


Figure 3.11: Result of the application developed - GoogleMaps map mode

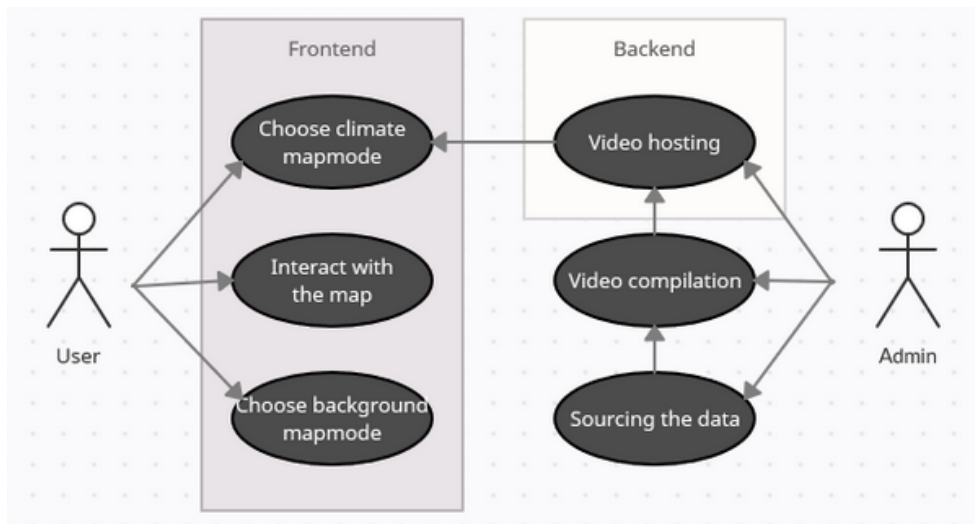


Figure 3.12: Current Use Case UML diagram

Chapter 4

Results and Future

4.1 Results

Upon analysing the current state-of-the-art, it is safe to conclude that the current popular GIS applications only display their data in the form of images layered on a map. This validates the main objective of the thesis. A GIS capable of overlaying video to a map has not been developed, so the creation of one has the potential to bring benefits to the user.

The source data used (the GIFs from Blitzortung.org) proved to occupy a bigger size than the compiled videos, as expected. The size of the source images is 14.2 megabytes, and the size of the output 64 videos is 8.7 megabytes, a 38.7% decrease in total size. Given that the GIF format already has some degree of compression, the value obtained is considerable. This also proves the thesis's primary objective.

Results tends to suggest that one of the major issue found the the current state of the art in website like earth.nullschool.net was mitigated with the prototype. In it's current form it's doesn't have any latency showing the information after the user refreshes page or interacts with the map. It is not possible to determine if this observation occurs cause of the new method of compiling the information to video and displaying or it occurs for the smaller amount of data presented.

Overall a process in which an administrator can manually source, compile and assign a set coordinates in order to present information was successfully developed. Although taxing for the actor in question, it is possible to develop and expand the backend of the system to be fully or partially automatic with assist of an human actor managing the system all the current manual steps. This is a strong basis that helps supporting the thesis's objectives, and with further development can improve the user's quality of life experience.

4.2 Future developments

The future of this thesis is to continue the development of the features that didn't manage to get implemented within the time limit set. It should prioritise the dynamic download and compilation of the data into video format and add more map mode options to the user, especially satellite-based information. A way to add control functionality like pause, play and rewind the video would also improve the quality of the user's experience. Lastly, compiling the videos on a format that handles an alpha layer that Leaflet can display and finding an alternative to imgur.com for video hosting, potentially hosting them on a private server. Fully automating the backend should also be a priority. In its current state a human supervising the system. The application should be able to source, recompile and correctly display the information in real-time according to the user's needs. This new system would only need the admin actor to manage and make sure it is working as intended.

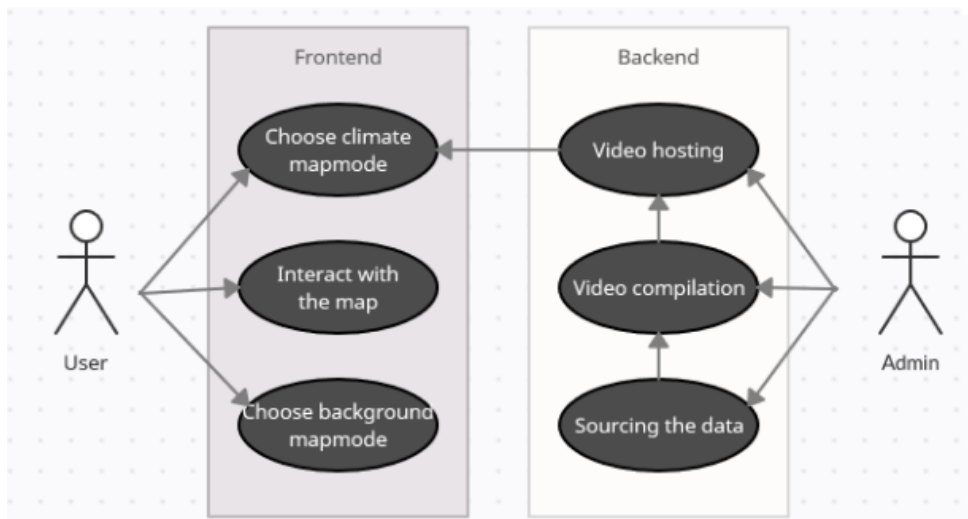


Figure 4.1: Fully automated Use Case UML diagram

References

- [1] Tor Bernhardsen. *Geographic Information Systems: An Introduction*. John Wiley & Sons.
- [2] K. K. Shukla and M. V. Prasad. *Lossy Image Compression: Domain Decomposition-Based Algorithms*. Springer Science & Business Media. Google-Books-ID: 77N26nH7u_wC.
- [3] Richard F. Haines and Sherry L. Chuang. The effects of video compression on acceptability of images for monitoring life sciences experiments, July 1992. URL: <https://ntrs.nasa.gov/citations/19920024689>.
- [4] Charles S. Swartz. *Understanding Digital Cinema: A Professional Handbook*. Taylor & Francis, 2005.
- [5] John Miano. *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*. Addison-Wesley Professional. Google-Books-ID: _nJLvY757dQC.
- [6] Thomas Boutell. PNG (Portable Network Graphics) Specification Version 1.0. Request for Comments RFC 2083, Internet Engineering Task Force, March 1997. URL: <https://datatracker.ietf.org/doc/rfc2083>, doi:10.17487/RFC2083.
- [7] Tiago Ferreira. GIF is (finally) dead. Long live Webp., July 2021. URL: <https://techdigitale.com/archives/1466>.
- [8] Iain E. Richardson. *Video Codec Design: Developing Image and Video Compression Systems*. John Wiley & Sons. Google-Books-ID: 8jxbbRKVbkIC.
- [9] Iain E. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons. Google-Books-ID: n9YVhx2zgz4C.
- [10] BoxCast Team. HEVC (h.265) vs. AVC (h.264): What's the difference? URL: <https://www.boxcast.com/blog/hevc-h.265-vs.-h.264-avc-whats-the-difference>.
- [11] Georgios Minopoulos, Vasileios A. Memos, Konstantinos E. Psannis, and Yutaka Ishibashi. Comparison of video codecs performance for real-time transmission. In *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, pages 110–114. doi:10.1109/ICCCI49374.2020.9145973.
- [12] Dan Grois, Tung Nguyen, and Detlev Marpe. Performance comparison of AV1, JEM, VP9, and HEVC encoders (conference presentation). page 120. doi:10.1117/12.2283428.
- [13] Jeffrey Ryan and Declan De Paor. Leveraging Earth and Planetary Datasets to Support Student Investigations in an Introductory Geoscience Course. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, pages EPSC2016–11575, April 2016.

- [14] Pierre Lassalle, Jordi Inglada, Julien Michel, Manuel Grizonnet, and Julien Malik. A scalable tile-based framework for region-merging segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 53(10):5473–5485, 2015. doi:10.1109/TGRS.2015.2422848.
- [15] Michael Dorman. *Introduction to Web Mapping*. CRC Press. Google-Books-ID: m6rMDwAAQBAJ.
- [16] Mark Monmonier. *How to Lie with Maps*. University of Chicago Press. Google-Books-ID: Q8OHiOiYwYUC.
- [17] John P. Snyder. Map projections: A working manual. Code Number: 1395 Code: Map projections: A working manual Publication Title: Map projections: A working manual Reporter: Map projections: A working manual Series: Professional Paper. URL: <http://pubs.er.usgs.gov/publication/pp1395>, doi:10.3133/pp1395.
- [18] Sarah Battersby, Michael Finn, E. Usery, and Kristina Yamamoto. Implications of web mercator and its use in online mapping. 49:85–101. doi:10.3138/carto.49.2.2313.
- [19] John P. Snyder. *Flattening the Earth: Two Thousand Years of Map Projections*. University of Chicago Press. Google-Books-ID: 0UzjTJ4w9yEC.
- [20] Official seal and emblem of the united nations : report of the secretary-general. page 4 p., Oct 1946. URL: <http://digitallibrary.un.org/record/552010>.
- [21] Time Inc. *LIFE*. Time Inc, Mar 1943. Life presents R. Buckminster Fuller's Dymaxion World; Google-Books-ID: WIEEAAAAMBAJ.