SediVis: Visualizing Sediment Plumes Dispersion

Luís Carlos de Campos Freitas

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor(s): Prof. Daniel Jorge Viegas Gonçalves
Prof. Telmo Morato

Examination Committee
Chairperson: Prof. Luís Manuel Antunes Veiga
Supervisor: Prof. Daniel Jorge Viegas Gonçalves
Member of the Committee: Prof. João Manuel Brisson Lopes

October 2018
Dedicated to my little niece, Constança
Acknowledgments

Over the past 6 years, my life changed in a way I will probably not be able to explain over the next few words. I will still try and give thanks to the people that helped me go through this.

Instituto Superior Técnico (IST) was always the one and only option for me. Back in 2012, I successfully made it to university. The beginning was hard. New schedule and harder topics to study. New people and more distractions than before. The first year and second year was when I learned resilience and endurance. Practice and more practice. But sometimes it seemed not to be enough. One thing is for sure, I developed a thick, thick skin. I cannot be less grateful to IST to, in some way, teaching me this.

IST has built amazing relationships with other universities around the globe. I was lucky enough to embark on a viking ship to Sweden on an ERASMUS programme back in 2015 with, the one and only, Alexandre Gordo. A huge thanks to KTH, Stockholm and, Alexandre Gordo for being a loyal friend over the past 5 years.

Have you heard of ERASMUS’ depression? Well, I am a living proof that it is true. At least it happened to me. It takes a while to recover, but it is only possible to do it with the right catalyst. For me, it was João Pestana. Thank you for the conversations about everything and nothing at the same time. For not judging me and for providing me the best vegetarian food.

Back on my feet and finishing the first year of my Master, and met again an amazing girl, Bárbara Bernardino, that I have met 2 years before in IST’s cantine. Gladly, this time she looked at me. Thank you for being my support, confident, and for giving me more than I have ever asked.

This was not possible without the amazing family I have. Thanks to my mother, father and brother, for being always there.

Thanks to Nuno Fontoura, Rui Santiago and João Esteves for going through many academic moments and more with me.

Fast forward to today, thanks to Prof. Manuela from IMAR/MARE for the long hours on Skype and the insightful information.

And last but not least, thanks to Prof. Daniel Gonçalves for giving me the chance to make this work, for helping me and for the hours spent around this project.

A lot have changed over the past 6 years, but the IST was always the constant. I cannot decide if that is a good thing, or a bad thing. You decide. Thank you for reading this.
Resumo

Nos últimos anos, o mar profundo foi reconhecido como uma fonte de metais raros. Com o aumento da preocupação com impacto ambiental das atividades de exploração miniera, foram desenvolvidos modelos de previsão que simulam a descarga de sedimentos, resultantes destas atividades. Os dados produzidos por estes modelos têm que ser visualizados de uma maneira apelativa — para alcançar o público-geral e entidades governamentais. Com isto, animações que demonstrem alterações espaciais ao longo do tempo são o método escolhido para disseminar esta informação. Nós propomos um sistema chamado Sedify. Este sistema permite visualizar os dados em três dimensões e observar as suas alterações ao longo do tempo. Também permite capturar fotografias de pontos relevantes na visualização, que posteriormente são utilizadas para a criação de vídeos através de mecanismos de interpolação. Foram efetuados testes em conjunto com profissionais desta área de investigação (IMAR/MARE e MARETEC) e Testes de Usabilidade. Os testes revelaram que este sistema é útil, tem uma interface simples e usável, mostrando desta maneira que é uma boa alternativa a soluções existentes.

Palavras-chave: exploração mineira, mar profundo, dispersão de sedimentos, visualização de informação, animações
Abstract

In recent years, the deep-sea was recognized as a source to scarce metals. With the rising concern on the impact of mineral exploration activities, predictive models have been developed to simulate the discharge of sediments, from mining operations. The data produced by these predictive models have to be visualized in an appealing way — to reach non-experts and decision-making entities. For this, animated videos showing spatial changes over the time are generally the chosen method used to disseminate this information. We propose a system called Sedify. This system allows data visualization in 3D and to observe changes over time. It also allows to capture snapshots of relevant points in space and time, that are used to create videos through interpolation mechanisms. This system was evaluated by experts from IMAR/MARE and MARETEC, and via Usability Tests. The tests revealed that this system is useful, with a simple and usable interface, proving to be a good alternative to existing solutions.

Keywords: Deep-sea mining, sediment dispersion, information visualization, animated videos
Contents

Acknowledgments ......................................................... v
Resumo .................................................................. vii
Abstract .................................................................. ix
List of Tables ................................................................. xiii
List of Figures ................................................................. xv

1 Introduction ................................................................. 1
  1.1 Objectives ............................................................ 2
  1.2 Contributions ......................................................... 3
  1.3 Thesis Outline ......................................................... 3

2 Related Work ............................................................... 5
  2.1 Sediment Plume Dispersion ........................................ 5
  2.2 Flow Visualization .................................................. 6
    2.2.1 Representation .................................................. 7
    2.2.2 Color ............................................................ 10
  2.3 Interaction ............................................................. 11
  2.4 Implementation considerations ................................... 13
  2.5 Online browser-based tools & animated videos ............... 14
  2.6 Discussion ........................................................... 17

3 Sediment Plume Dispersion ........................................... 21
  3.1 3D hydrodynamic modelling using MOHID water ............ 22
  3.2 Datasets ............................................................... 23
    3.2.1 Grid ............................................................ 23
    3.2.2 Results ........................................................ 24
    3.2.3 Time ........................................................... 24
  3.3 Requirements ......................................................... 25
    3.3.1 Current visualization pipeline .............................. 25

4 Sedify ................................................................. 27
  4.1 Architecture and System Overview .............................. 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Data Preprocessor</td>
<td>30</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Raw Data</td>
<td>30</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Preprocessing Raw Data</td>
<td>31</td>
</tr>
<tr>
<td>4.3</td>
<td>REST API</td>
<td>33</td>
</tr>
<tr>
<td>4.4</td>
<td>Sedify: First Prototype</td>
<td>35</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Data Visualization</td>
<td>35</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Concept</td>
<td>36</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Interface Prototyping and Data Loading</td>
<td>36</td>
</tr>
<tr>
<td>4.4.4</td>
<td>First Interface and Collecting Snapshots</td>
<td>37</td>
</tr>
<tr>
<td>4.4.5</td>
<td>Generating Videos</td>
<td>38</td>
</tr>
<tr>
<td>4.4.6</td>
<td>Improvements</td>
<td>40</td>
</tr>
<tr>
<td>4.4.7</td>
<td>Feedback</td>
<td>42</td>
</tr>
<tr>
<td>4.5</td>
<td>Sedify: Operational Prototype</td>
<td>42</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Improved Interface and Data Loading</td>
<td>43</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Creating and Exporting Videos</td>
<td>46</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Change options</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>Evaluation</td>
<td>55</td>
</tr>
<tr>
<td>5.1</td>
<td>Experts’ Feedback</td>
<td>55</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Methodology</td>
<td>55</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Results</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>Usability Tests</td>
<td>60</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Methodology</td>
<td>60</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Results</td>
<td>65</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>71</td>
</tr>
<tr>
<td>6.1</td>
<td>Achievements</td>
<td>72</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>72</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>75</td>
</tr>
<tr>
<td>A</td>
<td>Usability Testing Questionnaire</td>
<td>79</td>
</tr>
<tr>
<td>B</td>
<td>Datafiles Information</td>
<td>83</td>
</tr>
<tr>
<td>C</td>
<td>Results Tables and Charts</td>
<td>87</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Related work overview. ................................................................. 17

3.1 Areas submitted by Nautilus Minerals Inc. for deep-sea SMS mining exploration in the Azores region. ......................................................... 22

5.1 Experts questionnaire with median and IQR calculated from the responses. Minimum and maximum values also presented. ............................................. 58
5.2 Profiling questions with median and IQR calculated from the responses. ................................................. 66
5.3 Time and Success Rate for each task. .......................................................... 66
5.4 Experience related questions. ................................................................. 69
5.5 SUS Questionnaire ............................................................................. 70

C.1 Usability Testing. First two questions of the questionnaire. ................. 88
C.2 Experts questionnaire with median and IQR calculated from the responses. ................................................. 88
C.3 Tasks’s execution times with calculated IQR, Standard Deviation and Confidence Interval. 88
## List of Figures

2.1 RAIA’s Observatory monitoring network ........................................... 6
2.2 Flow representations. ......................................................................... 7
2.3 Pathlets (left) and streamlets (right) comparison on the same flow field [JRS12] ......................................................... 8
2.4 Same flow field with different representation methods [WBM+16]. ........ 9
2.5 The study of flows at different depths [BW11] .................................... 9
2.6 Sediments traveling through a shallow area ........................................ 10
2.7 The same map represented by different colormaps [SPG+15]. ............... 11
2.8 User interface example [RWW14]. .................................................... 12
2.9 HINT.FM Wind Map screens. ............................................................ 14
2.10 Two online browser-based tools screenshots. ..................................... 15
2.11 Videos’ screenshots. ......................................................................... 16
2.12 Uriel Zajaczkowski’s videos screenshots. .......................................... 17

3.1 Areas for SMS mining exploration in the Azores region. P is Patorra, M is Moreto, A is Arinto, F is Famous, S is Saldanha, and V is Verdelho. Image provided by IMAR/MARE team. ................................................................. 21
3.2 2D concentration visualization in Patorra and Moreto areas during one year simulation. Image provided by IMAR/MARE. ......................................................... 22

4.1 Sedify’s architecture. ......................................................................... 28
4.2 Frontend layers composed by WebGL and UI. .................................... 29
4.3 Spiral development methodology [Boe88] ........................................... 30
4.4 The contents and structure of a HDF file. .......................................... 31
4.5 Mapping a timestep to the corresponding date. ................................... 32
4.6 JSON file structure. ......................................................................... 33
4.7 Data structured to be used by the REST API. ...................................... 34
4.8 REST API request/response diagram. .............................................. 35
4.9 A concept of visualization: 3D plots example. ................................... 35
4.10 2D bathymetry array to 3D and 2D model of Patorra’s deep-sea terrain. The 2D array and the 2D model are both screenshots from the HDFView software. The 3D model is a screenshot from Sedify. ......................................................... 36
4.11 Up: Progress bar that indicates the current visualized date. Bottom: Collected frames to export as a video. ................................................................. 37
4.12 dat.GUI interface example. ............................................................... 38
4.13 Horizontal list of collected snapshots (camera roll). ........................... 38
4.14 Snapshot structure. ........................................................................ 39
4.15 First prototype’s main screen. ........................................................... 39
4.16 Intermediate frames calculation between Snapshot #1 and Snapshot #2. 40
4.17 A screenshot of the first video presented to the IMAR/MARE team. ........ 41
4.18 Operational prototype main screen. .................................................. 43
4.19 Multiple purpose bar. The Camera roll menu being presented. .......... 43
4.20 Color scale and visualization options. ................................................. 44
4.21 The interface elements used to interact with the time component of the application. 44
4.22 Comparison of three moving particles with a tail size of 10 units. A: Represents the current state of the particle. B: Represent all the ten past states. ................. 45
4.23 10 days of simulation of Patorra area, between the 16th of January and the 26th of January of 2011. Flow visualization example with a 5 unit tail size. ............... 46
4.24 Example of color schemes supported by the application. ................. 47
4.25 Taking a snapshot. Red box: Button that triggers the capture of the scene. Pink boxes: The captured scene and the preview in the list with the duration of the video until that frame (bottom-left). ................................................. 47
4.26 Snapshot screen. Is is shown the date, a picture of the scene, transition properties, delete and load buttons. ......................................................... 48
4.27 Drag&Drop snapshots feature. ........................................................... 49
4.28 Render screen. Before rendering the screen shows information about the video to be rendered. During the rendering process, a loading bar is presented. ............... 49
4.29 Rendering mechanism. Fluxogram for intermediate frames computation. 50
4.30 Camera motion types. Right: Rotation. Left: Translation. ....................... 51
4.31 Few examples of easing functions supported by the system. ............... 52
4.32 The users are able to watch and download all the created videos during the session. .... 53
4.33 The Options’ sub-menus. ................................................................. 54

5.1 Distribution of the experts’ answers plotted as a boxplot diagram. ....... 58
5.2 User experimenting the application. ..................................................... 61
5.3 The session presented during to the participants on Task 1. ................. 62
5.4 Video presented on Task 2. ............................................................... 63
5.5 Screenshot presented on Task 3. ......................................................... 64
5.6 Task 1 boxplots. .............................................................................. 66
5.7 Task 2 boxplots. .............................................................................. 67
5.8 Task 3 boxplots. .............................................................................. 67
5.9 Comparison of right and wrong for each task. ................................. 69
5.10 Distribution of the experience sentences’ evaluation, plotted as a boxplot diagram. 69
5.11 Distribution of the average SUS scores plotted as a boxplot diagram. .................. 70
C.1 Distribution of the SUS questionnaire, plotted as a boxplot diagram. .................. 87
Chapter 1

Introduction

The ocean is fundamental to life on Earth. Representing about 97% of our planet’s water [Nac67] and 70% of its surface [WSJS87], it is estimated that its deepest point can go up to at least ten thousand meters [Qiu17]. The ocean directly influences climate, weather and air quality, which then influence many other conditions of the utmost relevance for life on Earth. It has been well established that we know little about the ocean, its phenomena and the human impact on its ecosystem, particularly in the deep-sea [TR17]. Humans have been using the ocean for fishing activities for thousands of years, and lately, to explore the deep-sea for minerals. These activities not only have direct impact in the ocean’s ecosystem, but also on every form of life on Earth. It took many scientific research teams, in the past decades, to create tools to monitor the ocean, to collect large amounts of data and create simulation models to establish the baseline information needed to discover and better understand the ocean’s complex system [BBI16].

Oceanic and coastal observatories are networked infrastructures with sensors that collect data of physical, geological, chemical and biological variables in oceanic and coastal areas. They play a major role on monitoring and gathering data from the ocean, useful to understand its fundamental processes and the impact of human activities [BBI16]. The data, collected by sensors or derived from prediction models, is then visualized in multiple ways. The visualization can be made possible through tools to interact directly with the data or with videos that show the data over time, allowing the possibility to make public the information that relies beneath it. The representation needs to be appealing to get the public and decision maker’s interest, because, otherwise, it is either impossible to understand or not interesting enough to reach non-experts or governmental organizations.

In the past few years, the deep-sea was recognized as a possible source to scarce metals [PKA16]. The possibility of exploring the deep sea raises some concern on the impact on the marine environment, mostly because of the sediments released during and after exploration [BRCG13]. Azores’ area might be the only viable option for mineral deep sea exploration, being the only EU Member States EEZ with enough mineral reserves worth exploring. Nautilus Minerals Inc. has already applied for exploration in Azores [ECO14].

A scientific research team from Instituto do Mar (IMAR) and Marine Environmental Sciences Centre
(MARE) of the Departamento de Oceanografia e Pescas, University of Azores, developed a model to predict the potential dispersal of sediment plumes originated from the deep-sea mining operations. The data generated by IMAR/MARE’s research team, provides the possible sediment dispersion in three-dimensional high-resolution — 100x100km x 2km, during one year. The resolution is dynamic, being <1m near the surface and around 150m at 1000m at depth. Large datasets produced by simulations models or acquired by sensors, usually require data visualization techniques to extract valuable information. The goal of data visualization is to represent data in a readable way, to access and understand information easily [CEH+09].

Despite having the data resultant of running the predictive model, IMAR/MARE’s research team needs to visualize and present their data in an appealing way. In order to reach the general public and decision-making entities with a more perceivable approach, the research team needs a tool to interact with the data and keep the frames of important and relevant tracked events. This way, a video can be generated with the relevant information found. To achieve this, a system called Sedify, a browser-based tool, was developed. This system will help experts navigate in space and time through their data, and create videos that “tell” a story.

1.1 Objectives

For the success of this work, we believe that a simple yet effective goal should be defined. In this work, our goal is:

To develop a tool that visualizes sediment plume dispersion in the deep-sea and facilitates the creation of animated videos suitable for dissemination and decision-making.

In order to be successful, this system must satisfy the following key points:

- Allow the visualization of the sediment plume dispersion with color coding and flow representation techniques;
- Allow a free spatio-temporal exploration;
- Include pan, zoom and rotate camera controls;
- Allow the user to select relevant frames and collect them;
- Generate and export a video with the collected frames.
1.2 Contributions

The contributions of this work are:

- **Review of work related to the visualization of particle flows and ocean phenomena tools and videos** We have reviewed work done in the visualization of flows, interaction with three-dimensional scenes and spatiotemporal data visualization tools and videos, to better understand how to build and create a powerful tool.

- **Design and development of Sedify application** We built Sedify to help the visualization of particles flowing in the deep-sea. Users can visualize spatiotemporal data and create animations in a browser-based solution. The source code can be found at https://github.com/LewisFreitas/sedify.

- **Development of a data preprocessor** We developed ways to preprocess the raw data to be used by web systems. The repository can be consulted on https://github.com/LewisFreitas/sedify-data-preprocessor.

- **Usability evaluation of Sedify** We conducted usability tests of the interface of Sedify. The goal of this evaluation was to identify usability issues with the interface of the application and to determine if it meets the users’ expectations.

1.3 Thesis Outline

This document is organized the following way. Chapter 2 covers browser-based tools that are used to visualize meteorological conditions and ocean phenomena and animated videos generated by some of these tools. This section also covers work on flow visualization, interaction with tools and implementation concerns, followed by a discussion on the reviewed work. Chapter 3 gives a theoretical background on the sediments’ model and visualization is a section focused on the modelling work done by IMAR/MARE and the data resultant by their simulations. This section also focus on the structure of the data set and the tools used by the experts to extract, analyze and visualize the data. Chapter 4 provides an extensive explanation on the developed tool. This section focus on the software development decisions and data visualization concerns. Chapter 5 provides an explanation of the methodology used to evaluate our work and analysis on the results of the evaluation. Finally, Chapter 6 contains concludes the dissertation with the major achievements of this work and future work.
Chapter 2

Related Work

The public interest in marine issues and oceanic observatories, as well as the number of available visualization and interactive browser-based tools regarding ocean phenomena, has been growing in the past few years [ORVGNC14]. Nowadays, oceanic observatories can collect large amounts of data with sensors, that then analyze and turn it into information about not widely or deeply understood oceanic areas. The gathered information is helpful to many stakeholders, for example, it can be helpful to decision makers on how to keep populations safe and the general sustainability of marine ecosystems [Jon92].

The RAIA observatory, for example, is a multi-platform coastal and oceanic observatory, that is situated in the Northwestern Iberian Peninsula (Fig. 2.1). It was created due to the need of quality data and information of the coastal and oceanic areas in this particular Iberian location. This observatory has a broad range of available products and services that are used to monitor and collect data, generate numeric models of oceanic and meteorological phenomena and provide information to the population, scientists and decision makers [BBI16], such as, Alavai, a tool that visualizes predicted drift trajectories at the sea surface [BBI16, OBRV15].

2.1 Sediment Plume Dispersion

A sediment plume is a layer of large amounts of sediments suspended in water, and can be formed due to natural causes or human intervention. An example of human intervention is bottom trawling, a fishing technique known to cause dispersion and suspension of sediment plumes in water. It consists on pulling a fishing net across the seafloor and has been used for, at least, seven hundred years [Col89]. It is well established that it affects directly the environment by scraping and digging the substrate, dispersing suspended sediments, destroying marine benthic ecosystems and dumping processing wastes [RBGB+15].

Sediment resuspension due to trawling forms a suspended cloud of sediment that reduces the light levels of the substrate and, once it settles, it smothers the benthos. A study was conducted in a deep-sea environment at around 4000m and the effects were not fully reversed within 6 months, leading to the conclusion that the substrate would take decades to recover. The trawl gears can also vertically redistribute sediment layers, causing the mixing of surface materials into subsurface layers. This has direct impact
on the nutrient availability to filter feeders, a group of suspension feeding animals, making them filter more material with less food value, therefore less nutrients [FBCM11]. It can take the fauna decades to recover completely from the impact of this fishing technique [Jon92].

![Figure 2.1: RAIA’s Observatory monitoring network, located in Northwestern Iberian Peninsula [BBI16]](image)

Similar to bottom trawling, deep-sea mining and mineral exploration can have negative impact in the marine environment [BRCG13]. The vast negative impact of mining and mineral exploration in the sea has been widely attributed to the poor regulations and management in place on an international level. In this way, principles were proposed to guide the disposal of mining and mineral waste materials. These principles allow this disposal to be managed in a way that maintains its stability, meaning that the materials disposed should remain physically, chemically, geographically and radiologically stable over time, to limit its interaction with the ecosystem and to contain it in a small and confined geographical location [FBCM11].

The importance and relevance of visualizing the sediment plume dispersion is, either to minimize the impact on the ecosystem or to establish ways of conducting businesses in a field that is not well regulated. Information visualization can be helpful to detect and apply regulations beforehand. Marine logistics and maintenance of extensive and expensive marine engineering projects in oceanic and coastal areas require institutions to have and provide access to relevant information in an understandable way over the web [RWW14].

### 2.2 Flow Visualization

As previously mentioned, simulation models and sensors are currently generating large amounts of data. Flow visualization has been applied to meteorology and oceanology, to better represent flow phenomena that underlies in these complex, large and time-varying datasets. It has been a challenge for ocean visualization, the development of interactive visualization tools to represent flow fields. [WZAM09].

In the last couple of years, almost no effort has been made to develop of a good visual representation of ocean and wind, currents and patterns. The representation of vector fields, as winds, currents or
waves, have remained the same (i.e., grid patterns of small arrows or wind barbs) for more than 50 years [WKP14]. The visualization of continuous multivariate maps is done by many disciplines, such as oceanology and meteorology. Prior lessons learned for meteorology can be adapted to sediment transport visualization [EBM+13].

In order to discuss flow visualization, it might be useful to start by illustrating relevant concepts. A vector is represented by its **magnitude** and **direction**. Direction can then be categorized into orientation and vector sign, as show in Figure 2.2(a). **Pathline** is the line followed by a given particle in an unsteady flow. A **streamline** is the line that is tangent to the flow direction. A **streakline** is the line or the group of lines represented by the continuously connected particles emitted from a source point. **Streamlets** are short streamline segments, that show the direction of a traveling particle at any point in time. The size of these segments is proportional to the flow magnitude at its seed point. A **pathlet** is a short pathline segment that represents a single traveling particle (Fig. 2.2(b)).

![Representation of a vector](image1)

![Pathlets example](image2)

Figure 2.2: Flow representations.

### 2.2.1 Representation

Backed up by neuroscience and the theory of human perception [EBM+13], Ware et al. proposed an alternative representation of vector fields. To better represent the flow’s orientation, the design of a display should consider that the neurons that encode orientation should signal orientations that are tangential to the flow direction. When designing animated streamlines, magnitude is mapped to the speed of motion, direction shows both the orientation and the vector sign [WBM+16]. To represent the vector sign, head-to-tail asymmetry was chosen, such as arrowheads or teardrop streamlets [WKP14, WBM+16]. The speed was mapped to color or line thickness [WBM+16].

The study of ocean flow models require ocean flow patterns to be well represented [BW11]. Butkiewicz and Ware developed a solution to visualize flow patterns of traveling particles in the ocean. Their particle system consisted of pathlets. These pathlets convert particles’ position into pointers of speed and direction. They are represented as line segments that trace the path of the particles’ past positions, limited by a time threshold. The user has the ability to specify the elapsed-time to draw the pathlets, giving more or less visual weight, more or less historical trajectory data. The opacity is given by the ratio of the time elapsed from the current frame over the maximum elapsed time of the pathlet, providing a perceptual hint of the pathlet’s direction, even in static screenshots. The speed is represented by the vector’s magnitude. As particles move faster, their pathlets trace out a longer paths. This way, pathlets cover more pixels in total - faster flows will have more visual impact, while slower flows will have more
sparse pathlets. The pathlets’ depth representation will be discussed further on (subsection 2.6).

As mentioned above, pathlets show the history of particles’ trajectory - their traced path and where they are headed. This can have a negative impact in the visualization, since it does not provide any information of the current state of the flow field [JRS12]. Streamlets were chosen to represent sediment particles by Englert et al., in their approach. These streamlets represent the flow’s magnitude at any given position in time, as opposed to pathlets. The cause of strange behavior of streamlets was attributed to modeled numerical outliers. Streamlines are generally accepted to be the best choice when representing vector fields [WBM+16]. They are easy to implement and have intuitive semantics [Ira02].

Ware et al. provided an empirical comparison of static and animated representation and evaluation of simulated flow patterns [WBM+16]. Two experiments were conducted to determine which representation is the best choice at represent flow fields. In order to do this, the following visualization methods were tested: Animated streamlets, animated orthogonal particles, Static Arrow Grid and Static Equally spaced streamlines. In the first experiment, the participants had to detect a target pattern in a simple simulated flow field. The second experiment consisted of the participants giving an estimation of the path of an imaginary particle placed in a flow field. The authors’ hypothesis was that animated flow patterns would speed up visual searches for anomalous patterns and would improve advection pathway tracing. The results of the first experiment showed that animated flow patterns are the most effective way of showing flow direction. The second experiment showed that animated streamlets and Static Equally spaced streamlines are good choices. Even though animated vector field flow visualizations are easier to understand and the streamlets interfere less with background information than static visualizations, they are more expensive in computational terms. The visual cortex of the human brain processes moving and static patterns differently.

A human-in-the-loop optimization study was conducted, to determine the best way of representing dense patterns of Equally Spaced streamlines and streamlets. The participants, including specialists, had to tweak parameters to get the best representation possible of the flow currents. The parameters were changed through an interface that had 22 parameters mapped to data’s representation [WKP14].

Streaklines may resemble dyes, released gradually from a fixed location during time. Butkiewicz and Ware implemented a dye release tool in their system [BW11]. This turned possible the visualization of traveling particles continuously getting emitted in the user’s chosen location. The authors created a
vertical dye release pole that emits particles in different locations at depth. This was done using different colors for each depth range, turning possible the visualization of the difference in flows at various depths.

(a) Dyes: Particles getting emitted at different depths allowing the visualization of different flows at multiple depths.

(b) The dye pole editor. This editor allows the users to choose different colors for the released particles at multiple depths.

Figure 2.4: Same flow field with different representation methods [WBM16].

Figure 2.5: The study of flows at different depths [BW11]
2.2.2 Color

Multivariate maps usually have to deal with various representations at the same time. It is sometimes useful to take full advantage of multiple visual channels. The color channel is usually chosen, because color adapts very easily to different representations and scales very well on maps [WKP14]. Since color adapts very easily to different representations, it can be applied in many different contexts. For example, Englert et al. used color to distinguish regions of deposition and erosion, blue and red respectively, in their sediment transport visualization tool [EBM+13].

![Figure 2.6: Sediments traveling through a shallow area and getting eroded (red) and deposited (blue) [EBM+13].](image)

On a different context, blue and red were chosen by Resch et al. to represent shallow and deep areas, accordingly, in a 3 Dimensional (3D) time-varying context [RWW14]. Other different example is the Alavai’s drifting particles represented in black, that change to red and disappear once they reach the coastline [OBRV15]. As seen above, Butkiewicz and Ware used different colors to distinguish the emitting location at depth and the traveling pathlets color’s opacity was modeled by depth [BW11]. In deeper areas the pathlets are darker and more saturated than the ones near the surface.

One of the most difficult problems in designing effective flow visualization displays is dealing with the scale of the map. A great advantage of color coding values such as wind speed or surface temperature is that color tends to scale well. In a well-chosen color scheme, even if certain details cannot be seen, their colors will blend in the visual receptors to something approximating an average [WKP14]. Rainbow colormaps are misleading and obscure detail. Cool/warm colormaps are possibly the best broadly used solution, but usually the chosen colors are not effective. In a divergent colormap, white is placed at a point of interest, and two color spectrums diverge from that center point. Hue, Saturation and Value (HSV) are the color space correspondent to how people naturally experience color. In their study, HSV control points were equalized for perfection. It was decided to represent cool as green, because green has the largest perceptual range of any color. An asymmetrical divergent colormap was also considered by not placing the white in the center of the spectrum, while giving green a bigger percentage of the total space. This turned out to expose greater detail of the data and also for the chosen areas of interest. The effectiveness of the green/blue asymmetric colormap was tested and confirmed by user studies. Nested colormaps were created to maintain the level of detail of colormaps, but to avoid the drawbacks
of asymmetric colormaps [SPG+15]. Figure 2.7 provides a comparison of the same map represented by different colormaps.

(a) Cool/Warm colormap (blue/red).
(b) Cool/Warm (blue/green).
(c) Nested colormap.

Figure 2.7: The same map represented by different colormaps [SPG+15].

2.3 Interaction

One major weakness in modeling and simulating tools has been the poorly designed interface and interaction. Existing sediment transport visualization tools are very limited in terms of features and not intuitive enough to visualize results [EBM+13]. Englert et al. proposed a solution consisting of an analytical data visualization environment where users have the ability to explore and manipulate the data. They concluded that a better approach than the existing ones has to focus on collecting, manipulating and preparing data, generate a view or multiple views (e.g., 2 Dimensional (2D), 3D, camera perspective manipulation), ability to explore in space and time, update model parameters with every user interaction, a usable and perceivable human interface.

While most researchers are still using traditional 2D tools to visualize complex ocean flow models, one frame at a time, ocean flow simulations are generating complex multi-layer 3D ocean flow models [BW11]. Butkiewicz and Ware presented a system that combines a stereoscopic rendering display and multi-touch interaction. The stereoscopic rendering helps with drawing 3D structures and patterns, and the multi-touch interaction allows natural and efficient navigation and manipulation in a 3D environment. Even though this system is designed to take full advantage of a 3D stereoscopic multi-touch display, it works on an ordinary desktop (i.e., computer with mouse and keyboard) with regular pan, zoom and rotate controls.

Visualization in 3D is known to be extremely difficult and prone to visualization problems, such as occlusion, lack of direction or depth cues. Luckily, this has been widely studied in the field of Computation Fluid Dynamics (CFD), a branch of fluid mechanics that uses data to solve and analyze fluid flow
Iramee provided an interactive solution to visualize flow in a streamrunner, where problems such as occlusion and scene complexity rise [Ira02]. This solution avoids the common 3D visualization problems by allowing the user interactive control over the evolution of streamlines. The user can tweak time parameters, animation speed and streamlines appearance parameters, as well navigate with rotation and zooming controls.

![User interface example](image)

Figure 2.8: User interface example [RWW14].

With the rise of 3D models, and therefore the spatiotemporal analytical need, Resch et al developed a browser-based tool (Fig. 2.8) that provides the visualization of marine geological data for public use [RWW14]. Their approach focuses mainly on usability criteria, performance parameters, implementation effort and to offer spatial information that supports decision-making. With this browser-based solution, users are able to explore space and time. Simple and intuitive pan and zoom controls, resembling Google Earth’s controls, were the chosen navigation controls. The navigation bar was placed on the left side and the tools on the right side of the window in this system. Interacting, visualizing and understanding a time-varying 3D map requires a set of cognitive and perceptive skills, that have to be previously acquired by a person. To optimize the learning effect and the efficiency when viewing and interacting with such map, a dual coding method was considered. This method consisted of bringing together symbols and color coding to graphically represent variables. This avoided too much visual information and increased the efficiency of processing the information by the user. Navigation in two dimensions was included, since it is easier than the navigation in three dimensions. The drawbacks to using 3D to display real-world scenes are: definition; distortion (spherical perspective); topographic irregularities; hidden objects; scale-dependent feature presentation; or geometric inconsistencies [RWW14].

Other example of a browser-based application is Alavai. The main purpose of this tool is to visualize predicted drift trajectories of oil spills, harmful algae bloom or other hazardous substances, at the sea surface and trapped particles near the surface. Its User Interface (UI) offers the user the possibility to select the start date of the simulation. The user is also able to select the place wanted to start the trajectory. By hovering, the display shows the possible particles trajectory, the time it would take to get to the predicted place. When the user selects it, the particles start to move and their trajectory will be drawn. A timeline bar was included showing the current date, hours and minutes included [OBRV15].
2.4 Implementation considerations

The development of a flow visualization browser-based tool raises questions regarding the requirements for representing marine geo-data in a spatial and time-varying environment. An evaluation process is needed to decide which available tools are best suited for a given problem. Resch et al. determined the following set of requirements to develop a tool:

- Supported by web-browsers;
- Up-to-date;
- Simple development environment;
- Good performance;
- Portable;
- Integrate data;
- Use of geo-coordinates;
- Provides 3D visualization;
- Low to non-existent licences cost.

Java 3D, Silverlight, Stage3D, Unity and WebGL technologies were compared in the evaluation process. WebGL, a JavaScript (JS) library used to render 2D and 3D interactive graphics, was considered the most appropriate to use, since it is suitable for all requirements. The authors also included Three.js, a JS library used to render 3D objects, when developing the tool. Three.js turns the development environment less complicated, providing developers with an extensive library of already developed objects and features [RWW14].

Handling large amounts of data and rendering large spatial datasets often limit browser-based tools and visualizations to succeed. Caching mechanisms can be implemented to deal with large datasets, helping the pre-processing effort in a browser [RWW14].

Animated videos are a common method to visualize high-resolution models. It is the case of Model for Prediction Across Scales-Ocean (MAPS-Ocean), a high-resolution model (i.e., 5 to 20 km grid cells), used to predict and understand climate change. The simulation can run for multiple days to generate a video to represent this model. A video was presented featuring 1.4 million cells on the horizontal grid, varying in size from 10 to 30 meters and, 60 levels of depth on the vertical grid, varying from one meter near the surface to 200 meters at depth. Efficiency and the Input/Output (I/O) scalability have to be carefully thought out when developing tools to render and generate animated videos [JWHP+11].

J. Williams et al. used raycasting to reduce the computational power on a 3D visualization [JWHP+11]. Raycasting is a rendering method that traces rays from the viewpoint through pixel centers and, with the help of an algorithm it reduces the number of samples without affecting the visualization’s quality. It reduces computational power by providing implicit surface evaluation without the need of processing a high number of geometric operations.
2.5 Online browser-based tools & animated videos

Interactive maps of wind, weather and ocean conditions, available on the internet, are relevant to discuss. These browser-based tools have implemented a set of features worth explore and describe. This section consists on describing some of these found videos and tools.

Fernanda Viegas’s and Martin Wattenberg’s Wind Map\(^1\) is a 2D browser-based visualization map that provides a real-time prediction of the wind currents in the United States of America (USA). Although it was created as an artistic piece of art, it is used as a tool for many activities dependent of wind speed information. Streamlines were chosen to visualize the wind currents. The speed is determined by their density per 2D cell, the opacity of white color and traces’ traveled distance. Pan and zoom were the two navigation controls implemented to interact with this visual display. When the user hovers the mouse pointer over a specific area in the map, an information box is triggered with the area’s geographical coordinates and the corresponding wind speed.

![Image: Streamlines representing the wind flowing across the USA.](image)

![Image: Map legend - Wind speed.](image)

Figure 2.9: HINT.FM Wind Map screens.

An alternative, but with more features implemented, is an application developed by Cameron Bercario called Earth\(^2\). Earth is a 2D interactive visualization tool that provides a visual display inspired by HINT.FM Wind Map, the previous described tool. It is a multi-visualization tool with data fetched by multiple sources, used to visualize wind and ocean currents, ocean waves, particles and chemicals in the atmosphere and, temperature. This tool was developed with D3.js, WebGL and other JS frameworks. Pan and zoom were the navigation controls implemented to interact with this tool. Similar to HINT.FM Wind Map, when the user selects a point in the map, an information box is also triggered with the geographical coordinates and the corresponding visualization variable value, for example wind speed or particles concentration. The flow fields are represented by streamlines and the user has the ability to choose the map’s overlay, which is represented by a color scale. The user can also manipulate time by selecting a past date to start the animation, since it has the variables’ historical record.

NASA’s State Of The Ocean (SOTO)\(^3\) is a browser-based application developed by the Physical Oceanography Distributed Active Archive Center (PO.DAAC). SOTO is used to explore and analyze

\(^1\)http://hint.fm/wind/
\(^2\)https://earth.nullschool.net
\(^3\)http://podaac-tools.jpl.nasa.gov/soto/
physical oceanographic data. Similar to Earth and HINT.FM Wind Map, this tool provides navigation through pan and zoom controls, multiple data sources, historical record and colored informational overlays. It also offers control over the time, with the possibility to select the start and end dates, as well as step size (i.e., 1 day to 1 year) and animation speed (i.e., 0.5 to 10 fps). This tool also offers the possibility to share a view of the map, by generating a link with the parameters used to produce the current view.

(a) Cameron Beccarios’s Earth. It shows the overlay informational box, the filters and options.

(b) NASA’s State Of The Ocean (SOTO) showing sea surface temperature. The interface consists on the timeline bar with the time controls (pause, play and skip frame). The overlay menu shows the available dataset layers to import to the animation.

Figure 2.10: Two online browser-based tools screenshots.

Cameron Beccario’s Earth offers no possibility for users to export animated videos, but nevertheless there are Earth’s timelapse videos on the official Facebook page and YouTube channel. A video\(^4\) (Fig. 2.11(a)) available on the YouTube channel, shows the Pacific typhoon season as one year time-lapse of ocean wave height, from 28th of April of 2015 to 28th of April of 2016. This shows the possibility of an already built tool to export videos with specific parameters.

The Scientific Visualization Studio of NASA’s Goddard Space Flight Center (NASA/SVS) has created animated videos of earth’s phenomena over the years. Perpetual Ocean\(^5\) (Fig. 2.11(b)) is the example of an animated video showing ocean’s surface currents around the world over the course of two years. Two versions were created: a short and a long one. The long one is 20 minutes long at 30 frames per second. Although this video only shows surface currents, it is possible to visualize undersea bathymetry. This was done with a topographic and bathymetric exaggeration of 20x and 40x, respectively.

Coronal Mass Ejection and Ocean/Wind Circulation\(^6\) (Fig. 2.11(c)), winner of the first place in the Video category of the 2013 International Science and Engineering Visualization Challenge what, is another example created by NASA/SVS. This video shows an interesting 2D to 3D perspective transition while visualizing ocean currents by “flying” over the surface and then “diving” to show under the surface. The currents are represented as arrowhead streamlines.

\(^4\)https://www.youtube.com/watch?v=GfeJY8-YI7I
\(^5\)https://www.nasa.gov/topics/earth/features/perpetual-ocean.html
\(^6\)https://svs.gsfc.nasa.gov/11003
Other example\(^7\) from NASA/SVS, is a 3D representation of the atmosphere CO2 flux patterns. In this video, Carbon Dioxide concentration was mapped to color and it is possible to visualize it changing over time with altitude, during one year. In this animation, it is possible to visualize the occlusion introduced by the 3D visualization, representing a common limitation in similar situations. This was partially solved by rotating the map over the time, allowing the visualization of occluded data.

(a) Pacific typhoon season in Earth’s video screenshot.

(b) NASA’s Perpetual Ocean screenshot.

(c) Coronal Mass Ejection and Ocean Wind Circulation’s under the sea ocean flow visualization. A 3D representation showing streamlines at different depths.

Figure 2.11: Videos’ screenshots.

A collection of videos belonging to Uriel Zajaczkovski, a Physical Oceanography student and scientist at Scripps Institution of Oceanography, can be found on his personal page\(^8\) and YouTube channel. His videos are helpful since they represent changes over time of several parameters in the ocean in two, three or both options. Screen examples of Uriel’s video can be found in Figure 2.12 and visualization details in Table 2.1. Uriel’s videos are referred as Uriel-I\(^9\), Uriel-II\(^10\) and, Uriel-III\(^11\) in this document. The oceanographer uses Paraview\(^12\) for the creation of their animated videos.

ParaView is an open-source platform used for data analysis and visualization. The three examples present on Figure 2.12 are animations created using this software. A discussion on this tool is present on Section 2.6.

---

\(^7\)https://svs.gsfc.nasa.gov/12445
\(^8\)http://pordlabs.ucsd.edu/uriel/
\(^9\)https://youtu.be/FvvSTVYTHK0
\(^10\)https://youtu.be/ORe4UDUV1ju
\(^11\)https://youtu.be/OFK_XyXZhIY
\(^12\)https://paraview.org
Table 2.1: Related work overview.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Timeline</th>
<th>Navigation controls</th>
<th>Flow/Vis and others</th>
<th>Type</th>
<th>Other</th>
<th>Animations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Englert et al., 2013</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Streamlines</td>
<td>Color</td>
<td>No</td>
</tr>
<tr>
<td>Resch et al., 2014</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Streamlines</td>
<td>Color</td>
<td>No</td>
</tr>
<tr>
<td>Bulskis &amp; Ware, 2011</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Pathlines</td>
<td>Streamlines</td>
<td>Color</td>
</tr>
<tr>
<td>Itacondi, 2002</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Otero et al., 2015</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Ware et al., 2014</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>HINT/PM wind map</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Streamlines</td>
</tr>
<tr>
<td>Cameron Beccario's Earth</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Streamlines</td>
</tr>
<tr>
<td>NASA: State of the Ocean</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NASA: Perpetual Ocean</td>
<td>No</td>
<td>Yes</td>
<td>Video</td>
<td>Yes</td>
<td>No</td>
<td>Streamlines</td>
</tr>
<tr>
<td>NASA: Following Carbon...</td>
<td>Yes</td>
<td>No</td>
<td>Video</td>
<td>N/A</td>
<td>Color</td>
<td>Video</td>
</tr>
<tr>
<td>Uriel-I</td>
<td>Yes</td>
<td>No</td>
<td>Video</td>
<td>N/A</td>
<td>Color</td>
<td>Video</td>
</tr>
<tr>
<td>Uriel-II</td>
<td>Yes</td>
<td>Yes</td>
<td>Video</td>
<td>N/A</td>
<td>Color</td>
<td>Video</td>
</tr>
<tr>
<td>Uriel-III</td>
<td>Yes</td>
<td>Yes</td>
<td>Video</td>
<td>N/A</td>
<td>Color</td>
<td>Video</td>
</tr>
<tr>
<td>NASA: Coronal Mass...</td>
<td>Yes</td>
<td>Yes</td>
<td>Streamlines</td>
<td>N/A</td>
<td>Video</td>
<td></td>
</tr>
</tbody>
</table>

2.6 Discussion

As it was mentioned in the beginning of this Chapter, there is lack of information and tools to visualize the sediment plume dispersion in the deep-sea. Tools and visualization techniques that provide good visualizations, help reduce the impact on the ecosystem or to establish ways of conducting businesses in a field that is not well regulated. Very little research was done to try and solve this problem from an information visualization point of view, or to develop tools that facilitate the process. This section we will make an overview and discussion of the related work presented and tools evaluated.

Our work is focused on developing a tool to visualize sediment plume dispersion in the deep-sea and facilitate the creation of animated videos suitable for dissemination and decision-making. The result of this section analysis is reflected and summarized in the form of a comparison table in Figure 2.1. All the compared tools or videos consist on visualizing data changing over time.

When comparing the visualization scene provided by the reviewed work in regards of a 2D or 3D visualization, we came to the conclusion that only three tools and two videos, presented both perspectives in their work. All of the tools provided interactivity with the scene. All the online browser-based tools,
implemented only 2D visualization. On the other hand, all the videos described in the same subsection have the visualization presented in 3D. Uriel-II is the only video where two different perspectives are presented at the same time. In conclusion, interactivity and a visualization consisting of both 2D and 3D visualization are essential for a good exploratory tool.

In a time-varying context it is useful to take full advantage of time control features. These features offer extraordinary abilities when implemented, making possible the easy visual detection of data changes. A timeline bar provides information on elapsed time, start and end times, and allows the user to move back and forwards in time. Most of the implemented solutions with this bar allow the user to interact with the visualization with regular video controls: pause, play, go backwards and go forwards \( n \) frames (\( n \) represents the number of the skipping frames).

Since it is not possible to interact with the time in an already exported visualization, most videos present an always visible text element of the date of the visualized data or the elapsed time. These indicators are necessary as visual indicators of progress. A color legend of the variables represented are also available in all videos and most tools.

It is necessary to have navigation controls in an exploratory context. These controls help the user to navigate through the scene and change the perspective view. Navigation controls are implemented in all the interactive tools, but obviously not in the videos. The rotate control is implemented in all the tools that had 3D visualization, since it is necessary to overcome common limitations of 3D visualizations (e.g., occlusion). Pan and zooming controls are necessary in both 2D and 3D scenarios. These three controls combined, help the creation of 3D fly-through visualizations. These type of visualizations are generated with the use of panning, zooming and rotation controls to simulate flying around or through a scene.

For the visualization of flow (FlowVis), pathlines and streamlines are the most common choice when visualizing flow fields. Pathlines are mostly chosen to visualize drift trajectories [OBRV15, BW11]. Color is a commonly chosen visual channel, used mostly as a scale. This visual channel works well with flow visualization methods, for example, it can be mapped to the speed of travelling particles or the concentration of particles in the given area.

When developing applications for the web, it is important to understand the limits of browsers. The development has to focus on a set of requirements that make it possible to create a usable application. The 3D visualization on the web is possible due to the integration with WebGL and frameworks that make it easy to create data visualizations. There are mechanisms that help the integration of large datasets, such as caching and data pre-processing.

Exporting and/or sharing an animation consists of creating a video with the desired views, and then export and share it. Unfortunately, only two tools reviewed in this section had this feature. NASA’s SOTO presented the possibility of generating a link with the created visualization. This link contained the parameters used to generated the animation. Upon opening the link, the browser will replicate the animation by requesting a visualization with the same parameters. On the other hand, ParaView, the tool used by Uriel Zajaczkovski, is a complete tool for both data visualization and analysis. This tool was not analyzed, but the work done with this tool was. This tool is limited in terms of 3D animations rendering quality and oftenly third-party tools are used for rendering. Besides that, ParaView has a steep learning
curve due to the extensive list of implemented features, and complex user interface and interaction.

In conclusion, this work could be helpful by bringing a way to visualize the data and provide means of interaction to create animations. It is relevant to understand that effort on breaking down complex visualization techniques into simple and usable application was researched. This research may be helpful to fulfill the lack of research done in the usability of software used to create spatiotemporal visualizations for the ocean research domain.
Chapter 3

Sediment Plume Dispersion

Since 2006, the government of Azores has been contracted by international companies with the intention of exploring mineral resources in the deep-sea. By 2012, the legislation for the mineral exploration and exploitation was created [ECO14].

Nautilus Minerals Inc. were the first company submitting a proposal for exploration rights in several areas around Azores: Patorra, Moreto, Arinto, Famous, Saldanha, and Verdelho. IMAR/MARE developed methods to simulate the potential dispersal of sediment plumes in two case study areas: Moreto (1000m depth) and Patorra (2300m depth).

![Figure 3.1: Areas for SMS mining exploration in the Azores region. P is Patorra, M is Moreto, A is Arinto, F is Famous, S is Saldanha, and V is Verdelho. Image provided by IMAR/MARE team.](image-url)
<table>
<thead>
<tr>
<th>Area name</th>
<th>Total area (km$^2$)</th>
<th>Location in EEZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arinto</td>
<td>1,564</td>
<td>Inside</td>
</tr>
<tr>
<td>Famous</td>
<td>1,515</td>
<td>Inside</td>
</tr>
<tr>
<td>Moreto</td>
<td>1,543</td>
<td>Inside</td>
</tr>
<tr>
<td>Patorra</td>
<td>1,608</td>
<td>Inside</td>
</tr>
<tr>
<td>Saldanha</td>
<td>1,544</td>
<td>72% inside</td>
</tr>
<tr>
<td>Verdelho</td>
<td>1,498</td>
<td>Outside</td>
</tr>
<tr>
<td>Total</td>
<td>9,272</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Areas submitted by Nautilus Minerals Inc. for deep-sea SMS mining exploration in the Azores region.

### 3.1 3D hydrodynamic modelling using MOHID water

MOHID water is a three-dimensional water modelling simulator developed by the Marine and Environmental Technology Research Center (MARETEC) at Instituto Superior Técnico, University of Lisbon [IPCST+13]. This tool has been widely used to simulate complex vertical and horizontal ocean circulation patterns and velocity fields, sediments transport and plume dispersion, among others.

This system allows to track the trajectories of selected water masses or particles. It was used to simulate the movement of the plume generated by deep-sea mining activities using transport fields calculated with the hydrodynamic model. The dispersion, the plumes and particles were computed to simulate the deep-sea mining activities at a discharge point.

Two different sediment plumes can be modelled: *in situ* and dewatering plume. *In situ* sediment plumes are originated by in situ mining operations. These plumes are usually generated when the Seafloor Mining Tool (SMT) lands or crawl on the seafloor. The sediments get suspended and dispersed by local currents. Dewatering plumes are generated when the water that contain particles is delivered back to the water at a discharge point.

Figure 3.2: 2D concentration visualization in Patorra and Moreto areas during one year simulation. Image provided by IMAR/MARE.
3.2 Datasets

The models described below output data that expresses the spatial changes over time, and other phenomena. There are data for all the six different locations, consisting of a high-resolution data model. The data is represented by:

- Total area: 100x100 km
- Cell area: 1.2x1.2 km
- Depth: 50 layer of cells
- Number of cells: Approximately 780,000
- Timestep: 60 seconds
- Number of timesteps: Approximately 525,000 (1 year of simulation)
- Visualization: Sediments plume dispersion trajectories and concentration of contaminated water, in 3D and over the time

The datasets are in HDF (Hierarchical Data Format) file format, more specifically in HDF5\(^1\). HDF5 is a data model, library, and file format for storing and managing data. This format is oftenly used for storing scientific data, because of its flexible and efficient I/O for high volume and complex data. It is the standard input/output format of temporal and/or spatial data sets in MOHID.

The files are composed by three main groups: Grid, Results and Time. Each group and their sub-groups are explained below in Section 3.2.1.

3.2.1 Grid

This field contains all the information regarding the geometry of the model. This group contains the following sub-groups: Bathymetry, Longitude, Latitude, ConnectionX, ConnectionY, VerticalZ and WaterPoints3D.

- **Bathymetry** Two dimensional array that expresses the depth of the underwater terrain. The unit of measurement is meters.

- **Longitude and Latitude** Two dimensional array that expresses the horizontal geometry of the model. The unit of measurement is decimal degrees.

- **ConnectionX and ConnectionY** Easting and Northing, expressed in meters (UTM).

- **VerticalZ** Vertical geometry of the model, compose by a certain amount of layers (i.e., 50 vertical layers). For each time instance, the model's layers can change in depth for each cell of the horizontal grid.

- **WaterPoints3D** 3D matrix that expresses for every cell if there is water in a cell (1) or not (0).

\(^1\)https://support.hdfgroup.org/HDF5/
3.2.2 Results

This group consists on the results in spatial data over time of the simulations. It contains as many subgroups as the number of emission points.

Every 1D array position is mapped to a respective particle group. For example, the first position of the Age array refers to the same particle group as the first position of the Latitude array, and so on. Some of the fields this group contains are described below.

- **Age** A 1D array that expresses the age in days. Each position refers to a different particle group. The length of this array is the number of existing particles at a given timestep. The unit of measurement is days.

- **Deposition State** A 1D array that expresses if a particle has been deposited or not at a given timestep. The length of this array is the number of existing particles at a given timestep. There are two possible values: 1 (has been deposited) and 2 (has not been deposited).

- **Latitude and Longitude** A 1D array that expresses the latitude or longitude of any existing particle at a given timestep. The unit of measurement is degrees.

- **Origin ID** A 1D array that expresses the ID of the origin of the particle, if many origins are present in the model. The length of the array is the same as the existing particles at any given timestep.

- **Volume** A 1D array that represents the volume of a particle at a given timestep. The unit of measurement is cubic meters.

- **X, Y and Z Pos** A 1D array that represents the position, in meters, of a particle at a given timestep.

- **Sediment** A 1D array that represents the concentration of a particle at a given timestep. The unit of measurement is milligrams per litre (mg/l).

- **Salinity** A 1D array that represents the salinity of a particle at a given timestep. The unit of measurement is milligrams per litre (mg/l).

- **Temperature** A 1D array that represents the temperature of a particle at a given timestep. The unit of measurement is degrees Celsius.

3.2.3 Time

Information of the date for every timestep of the model. Each Time field contains an array with the year, month, day, hour, minutes and second. The year can be accessed in the first position of the array, the year the second position of the array and so forth. This is useful to map the changes in the results to the time they occurred.
3.3 Requirements

Upon receiving data samples from the IMAR/MARE team, we kept having multiple informal meetings over time. These meetings helped mitigate questions about the data and to better understand how to build a system to properly visualize and generate animations with the data. Generally, these meetings were made over video-calls for 2 hours maximum or over e-mail with follow-up questions.

These meetings resulted on a collection of the following requirements:

- The system has to be easy to learn how to use;
- The system needs a user-friendly interface and experience;
- The system needs to allow free spatiotemporal data visualization;
- The system has to export a video of the changes over time;
- The system has to allow the user to visualize different variables and change visualization parameters.

3.3.1 Current visualization pipeline

The Chapter 2 presents work done in multiple fields of ocean phenomena visualization, sediments and particles dispersion and, mostly, the considerations when building such systems. With the help of an expert of sediment plume dispersion from MARETEC, we tried to understand how the pipeline of visualization could be implemented. After being generated by the water modelling system, MOHID water in this case, the data has to be somehow transformed to be visualized. While it is possible to visualize the data directly on the software, the user experience is really sacrificed and prone to be difficult. MOHID water has limited visualization features.

We learned how to transform the data to be visualized by other tools with scripts that transform data into other file formats. File formats can be either JSON or XML. After preprocessing the data, these datasets are ready to be imported by an external tool. Such as ParaView (also mentioned in Section 2.5 and 2.6), but the process is rather complex. To achieve better animation results, we might use Blender\(^2\), a well known 3D modelling and animation rendering package. This process offers good results, but it is a complex method, that has a steep learning curve, according to the MARETEC expert.

In conclusion, we believe that understanding this allows us to develop a solution to improve the current visualization pipeline. The next chapter explains in detail how we built our solution.

\(^2\)https://www.blender.org/
Chapter 4

Sedify

The analysis of existing tools and animations created for similar use cases, revealed a complex way of creating animations of spatio-temporal data, lack of a good user experience and/or a steep learning curve. To address this need, we believed that a minimalist, easy to setup with an improved user experience was in need.

Sedify is an easy to setup, browser-based cross-platform application that allows the visual exploration of sediment plume dispersion in the deep-sea and, the creation of animated videos, suitable for dissemination and decision-making. This tool was developed to help IMAR/MARE scientific research team to visualize data and produce videos, that might be useful for dissemination in political forums. We focused on maintaining the interface and experience as minimalist as possible, while providing important features for the activity of information visualization and exporting animations. This chapter focuses on the decisions and considerations made during planning and development.

We believe that this system’s success relies on well-defined requirements. We decided simple, yet crucial requirements to meet the experts’ needs. Our system:

- Allows the visualization of the sediment plume dispersion with color coding and flow representation techniques;
- Allows a free spatio-temporal exploration;
- Includes pan, zoom and rotate camera controls;
- Allows the user to select relevant frames and collect them;
- Generates and exports video containing the collected frames.

4.1 Architecture and System Overview

This system follows a common client-server architecture, composed by three major components: the frontend (client-side), the backend (server-side) that, under HTTP protocol delivers HTML and JS scripts.
and the data for the application, and a database. The architecture of this system can be seen on Figure 4.1.

For the development of this tool we focused on web technologies, using JS libraries. We have found that transmitting to and loading data on the frontend can become a very heavy process. If the size of data increases, the browser (frontend) will probably not keep up with the load. In a client-server architecture, it is usual to separate data operations (storage or preprocessing) and the user interface. This is achieved by locating these operations on the backend (Fig. 4.1).

Figure 4.1: Sedify’s architecture.

The frontend consists of a single-page application that requests data to display to the backend. The communication between both ends is made through a Representational State Transfer (REST), Application Programming Interface (API) with Hypertext Transfer Protocol (HTTP) requests. The API defines a set of endpoints, accessible through HTTP requests, resulting in a response containing data in JavaScript Object Notation (JSON) file format. The backend is responsible for handling data operations, consisting in preprocessing raw data to convert and deliver it in more widely accepted formats (e.g., JSON).

We have decided to use Node.js\(^1\) to develop our server, since it is a JS an open-source and cross-platform environment. Node.js provides a package manager, Node Package Manager\(^2\) (npm), one of the largest repositories of open-source JS libraries. We wanted to keep the development mostly on JS, because code can possibly be reusable on both ends. Our server takes full responsibility for the application’s data by fulfilling two main functions: preprocessing the raw datasets and making the proprocessed data available through the REST API.

The frontend consists on the visual part of the application that users can interact with. As mentioned earlier, our solution consists on a browser-based application. It was mainly developed in HTML, CSS and JS. This stack of the application is mostly responsible for taking care of the graphical interface and interaction with data. JS was chosen because it is supported by the most common modern web-

\(^1\)https://nodejs.org
\(^2\)https://npmjs.com
browsers. JS libraries were used to help with the development. These libraries were required to be open-source, up-to-date and with reduced integration complexity. Most libraries had documentation and integration tutorials available. For the interface design and development we mainly used Bootstrap\(^3\), which is an open-source toolkit that provides easy to design and develop interfaces through a responsive grid system and custom components. The icon toolkit used was by FontAwesome\(^4\), a large icon library.

This application's display has two layers (Fig. 4.2): the WebGL\(^5\) scene and the interface. It was developed to be interacted with mouse and keyboard. The mouse can be used to control the interaction with the WebGL scene and the interface, while the keyboard is mostly used to fill values in input fields. WebGL was chosen to render the 3D graphics in our application. It was the best option because of its support by most modern web-browsers and being up-to-date. To make development less complicated, Three.js\(^6\) was considered a good library to use, providing an extensive library of already developed objects, features and WebGL interface.

![Figure 4.2: Frontend layers composed by WebGL and UI.](image)

Our methodology approach to build this system was the spiral development methodology (Fig. 4.3), from which resulted the development of two prototypes. This methodology was chosen because software is produced early in the software life cycle and facilitates the need to change when researching and exploring new features or changes to prior developed features; thus reducing the overall risk [Boe88]. The results of this development methodology were the first prototype, described in Section 4.4, and the operational prototype, described in Section 4.5.

\(^3\)https://getbootstrap.com/  
\(^4\)https://fontawesome.com  
\(^5\)https://www.khronos.org  
\(^6\)https://threejs.org
4.2 Data Preprocessor

A data preprocessor was created to transform the raw data, provided by IMAR/MARE in HDF5 format, into a more widely used format. This helped reduce the original data size by selecting the relevant data for visualization and analysis. It was developed using Python. Once the data is preprocessed, it is stored in the server and ready to be served by the backend to the frontend application. This section covers an explanation on how the data was preprocessed.

4.2.1 Raw Data

The content of the datasets provided by IMAR/MARE were in HDF5 file format (.hdf5 extension), as introduced in Section 3.2. This format is known to be helpful when dealing with large and complex data collections. The files are structured hierarchically (Fig. 4.4(a)), allowing multiple datasets to be present in a single file. This format was not intended to be used by web applications, therefore its integration in web systems is not particularly easy.

At first, we managed to visualize the datasets with a software called HDF View\(^7\). This software allowed to visualize the file hierarchy in a tree structure (Fig. 4.4(a)) and to view the content of each dataset (Fig. 4.4(b)).

The visualization of the files’ content, led us to plan how to extract the relevant data for the application.

\(^7\)https://www.hdfgroup.org/downloads/hdfview
Due to the fact that the frontend requests data from the server, we had to plan for efficiency. For this we thought of ways of explicitly dividing parts of the data, creating subsets of each dataset. We also had to figure out a more appropriate data file format to accomplish the best efficient process possible. In the next section, we explain what was done to achieve this.

4.2.2 Preprocessing Raw Data

After visualizing the contents of the raw data, it was decided that it was possible to simplify the data by dividing it into the following datasets: Deep-sea terrain, Sediments and Date. Irrelevant data fields were ignored, allowing us to reduce the data size. We considered irrelevant those fields that would not add anything specific to the visualization and/or fields that the experts, that provided us the data, requested
us to avoid.

The Deep-sea Terrain dataset corresponds to the geometry of the deep-sea model, which is the grid mesh of the deep-sea terrain and its geographic coordinates information. From the raw data, we kept fields Bathymetry, Longitude and Latitude. Each field is a 2D array. The $n$-index of each array is mapped to the same cell in the 3D model. Bathymetry contains data relative to the ocean’s depth relative to the sea level for each Longitude and Latitude coordinates pair. In conclusion, the terrain grid dataset is composed of three 2D arrays. In order to access the information of the first cell of the grid, one has to access the first position of the three arrays, for example.

For the Sediments dataset, it was important to select what was relevant from the Results group. For the purpose of the visualization, we have decided to keep the data that is used to define the particles’ spatial position in the 3D model, that is the latitude, longitude and depth. These three fields can be mapped to the model coordinates (X,Y and Z). For a more appealing and informational visualization, the particles can be represented by other data fields. From the provided data, IMAR/MARE team decided to visualize concentration, temperature and salinity. In conclusion, this dataset is composed by the following datasets: depth, latitude, longitude, salinity, concentration and temperature.

The last dataset considered was the Date. This dataset is not only used to coordinate and map a unique timestep to the corresponding date of simulation, but also to map the date to the particles state in all the visualization data fields. This field contains information about the year, month, day, hour, minutes and seconds, for every timestep. The Figure 4.5 allows to better understand this process.

To extract the data, a program was developed to read and manipulate the raw data files. Our research led us to h5py\(^8\), a Python library that provides an interface to interact with HDF5 files, allowing the iteration through the original files. The first developed script, not only iterated through the datasets, but also extracted relevant data and exported it as binary files. A binary file, was exported for each property, for each timestep. For example, for the first timestep of simulation, we had a file for the latitude, one file for the longitude, one file for the salinity, and so on. The same applied to the deep-sea terrain and date.

This is a time consuming process, since it has to access the results for each timestep. For example, one year of simulation can have around 3000 timesteps. There are multiple timesteps that have hundreds of thousands of particles. Every particle has multiple properties, like concentration, latitude, longitude, depth, and so on.

Figure 4.5: Mapping a timestep to the corresponding date.

\(^8\)http://www.h5py.org/
depth, and so on. The binary files were used on the first iteration of the prototype (Section 4.4).

After that, we believed that this program could improve substantially, and decided to choose a better way to represent the data, while reducing the number of files. Two of the most widely adopted formats on the web are JSON and Extensible Markup Language (XML). We opted to use JSON, due to its simpler syntax and better integration with JS. This allowed to maintain a readable and easier to use data structure, when compared to having multiple binary files. It also allows to better integrate with our web system.

```json
{
  "sedify": [
    {
      "timestep": 1,
      "date": {},
      "sediments": {
        "concentration": {
          "unit": "mg/l",
          "data": []
        },
        "temperature": {},
        "salinity": {},
        "latitude": {},
        "longitude": {},
        "zpos": {}
      }
    }
  ]
}
```

Figure 4.6: JSON file structure.

At this point the script prepared the datasets to be used by the frontend application, grouping multiple data fields into JSON files (Fig. 4.6). It was possible to determine and select which data fields to include when exporting the files. This helped reduce data size. The data is stored in the server with the structure seen in Figure 4.7. This data is consumed by the REST API, explained next.

### 4.3 REST API

It is not feasible to load large datasets on a browser. We had to figure out a way to transfer efficiently smaller parts of data from the server to the frontend. This method would have to allow the request of specific parts of the dataset on demand. Our research determined that building a REST API was the solution.

An API is considered a set of procedures that perform one or more tasks with the intent of being used by other software. RESTful services are web-services that adhere to the REST architectural style. These systems are used to facilitate the communication of standards between systems on the web. We have decided to develop a REST API, because it allows a simple and uniform interface, and it is scalable.
This API may serve for this application, but it can be used to build any other systems with similar data logic. The REST API was developed using Express\(^9\). This npm package allows to create a server that answers to HTTP requests made to defined endpoints.

Having the data preprocessed, we proceeded to organize it into sub-groups. Figure 4.7 shows these system’s data sub-groups. The three groups are, as seen before, the Deep-sea terrain, the Date and the Sediments. The latter is subdivided into smaller sets. Each one of these smaller sets correspond to all the particles’ information for a given timestep (the number that identifies the subdivision), presented in Figure 4.6.

![Figure 4.7: Data structured to be used by the REST API.](image)

Once the frontend application requests data for a given timestep, the server interprets the request, and serves the requested data as a JSON file. This communication is done via HTTP protocol; the client requests the data with the GET method. Figure 4.7 presents the structure of the dataset served by the REST API. The frontend can request data from the server by calling the API endpoints. The small boxes inside Particles contain all the data (e.g. concentration, position, and so on.) about the set of particles at a certain point in time, grouped by the timestep. We have defined the following endpoints:

- **GET /api/sediments/**\(\{\text{filename}\}\)/\(\{\text{timestep}\}\)** Returns the particles’ properties for a given timestep and filename. The results contain time information, particles’ positions and other fields (e.g., concentration, temperature, ...).

- **GET /api/time/**\(\{\text{filename}\}\)** Returns the whole time information for a given filename. The results contain the year, month, day, hour and seconds, for all timesteps.

- **GET /api/terrain/**\(\{\text{filename}\}\)** Returns the terrain grid for a given filename. The results contain three arrays: bathymetry, latitude and longitude.

The filename is generally the name of the area of simulation or the when it was created (e.g. “Patorra-10-02-2018”). The timestep identifies the specific point in time of simulation. All the requests done over

\(^9\)https://expressjs.com/
the endpoints defined above, will trigger a response in JSON (Fig. 4.8).

![REST API request/response diagram](image)

Figure 4.8: REST API request/response diagram.

### 4.4 Sedify: First Prototype

One of the goals of our work was to create a simple and easy to use interface that provided all the necessary features to export animations successfully. The first prototype (Fig. 4.15) served as a way to present to IMAR/MARE research team the concept and how they could benefit from it. The results of this stage were a functional prototype with a simple interactable interface and a video partially generated with the prototype.

#### 4.4.1 Data Visualization

During this stage, we had to decide on how to visualize the data in an appealing and interactive way. We had two main structures to visualize: the deep-sea terrain and the particles. The deep-sea terrain is constant for a given dataset, but the particles’ their properties may change over time.

![Visualization examples](image)

(a) 3D scatter plot example.  
(b) 3D surface plot example.

Figure 4.9: A concept of visualization: 3D plots example.

The deep-sea terrain was decided to be represented as a 3D surface plot (Fig. 4.9(b)). The grid is created through mapping the depth on the Z-axis, the latitude to the Y-axis and longitude to the X-axis (Fig. 4.10). We decided to represent the particle system as a 3D scatter plot (Fig.4.9(a)). The particles are placed in the model by mapping their depth, latitude and longitude to the model coordinates (X,Y and Z). This allows not only to change them spatially, but also to apply colors to each particle for a given property (color coding). The end result is a combination of both plots.
4.4.2 Concept

Taking into consideration the requirements we had for this application, we sketched out a concept for the interface and interaction. Many alternatives were considered and iterated through until we decided on an interface to start with.

Our solution would need to implement free navigation, both in time and space. To freely navigate in space through the 3D scene, we chose the basic controls: rotate, zoom and pan. The mouse is used to perform these controls. To navigate in time, we sketched a slider progress bar with a visual indicator of the current visualized date (Fig. 4.11).

In order to allow the user to create animations, we thought of a slider bar that would represent the video to be exported. This bar would include the pictures/snapshots taken from the 3D scene, that would then be transformed into an animation. These pictures represented the sequence of frames of the final video the user would be able to generate. For the interaction with this bar, the user would be able to delete and organize the snapshots as pleased (Fig. 4.11).

The data would be transformed and visualized in a 3D scene, as seen in the Section 4.4.1. This prototype would allow the visualization of the deep-sea terrain and the particles.

4.4.3 Interface Prototyping and Data Loading

As soon as we started the development, we used dat.GUI\(^\text{10}\), a lightweight controller library for JS. This library allowed us to quickly create a simplistic User Interface (UI) with easily customizable buttons and parameter inputs. This was useful for developing and testing multiple parts of the software, and to

\(^{10}\text{https://github.com/dataarts/dat.gui} \)
understand the limits of this application without compromising development efforts. An example of its usage can be seen on Figure 4.12.

At this point, we did not have the backend fully developed, so we focused on building a fast to develop mechanism to load the data on the frontend. In order to do this, we developed a Python script to export specific parts of the datasets as binary files (e.g., the bathymetry or the particles’ position). These binary files were very small in size when compared to the original data, so it was possible to load them on the frontend. After loading the data, the frontend had to translate the data into JS objects. These objects corresponded to the deep-sea terrain and the particles.

Once the data was loaded on the client, we created the 3D model of the deep-sea and the particles using Three.js, that rendered both in a WebGL scene. The bathymetry data was used to create a plane (PlaneGeometry object) for the deep-sea terrain grid. This shape offered the possibility to modify the Z-axis through its vertices. The sediments spatial data (depth, latitude and longitude) was used to create points (Points object) in the 3D model. During this stage, we were also loading the sediments’ concentration data. It was used to map the sediments’ concentration to the points’ color, at a very early stage of the development.

At this stage, we had also developed features to allow the user to take snapshots of the scene. At this point, we did not have an interface to visualize those snapshots or interact with them.

4.4.4 First Interface and Collecting Snapshots

After building a solution for representing the data in a 3D scene, we started the development of the first interface. This interface would present features that were previously implemented using the dat.GUI interface. The UI offered the following features: take snapshots of the current visualization, view the current date of visualization and watch the collection of the snapshots taken. We tried to provide a general idea of how the elements played together, and where they would be located in the final interface.

The UI had a bottom bar where the snapshots taken would appear horizontally (Fig. 4.13), resembling a camera roll. We placed an always visible button on top of the bottom bar, that allowed the user to take snapshots of the current visualized scene. The snapshot taken would then appear on the camera
Figure 4.12: dat.GUI interface example.

Figure 4.13: Horizontal list of collected snapshots (camera roll).

roll. The bar also offered the possibility to select one out of three menus, to allow the interface to accommodate more features in separate menus (e.g., visualization parameters or scene settings). On the top-right screen corner, we placed a text container that showed the current visualized date. Note that the horizontal list of snapshots (camera roll) did not have any other particular feature other than offering a visual cue of the snapshots taken during the session. The state of this stage’s final interface can be seen on Figure 4.15.

When capturing a snapshot, it was relevant to store the camera’s position and focus point. The camera’s position consists of a triplet of numbers (X,Y and Z) that serve to identify the camera’s position in the 3D space. The cameras’ focus point consists of a triplet of numbers (X,Y and Z) that serve to identify the point in the 3D space, that the camera is orbiting around. The Figure 4.14 shows a representation of the snapshot structure. Note that a snapshot can store more information, but at this stage of development this was all that was stored.

In conclusion, we had followed the principle described in the Section 4.4.2. The camera roll followed the principle of the bottom bar in Figure 4.11. The date text container followed the principle of the upper bar in Figure 4.11.

### 4.4.5 Generating Videos

One of the features developed at this stage was the video rendering mechanism. From the collection of captured snapshots it was possible to generate the in-between frames of consecutive snapshots, to generate an animated video.
The generation of the transition frames, the in-between snapshots’ frames, consisted of interpolating through all the steps in-between a starting point and the end point. Interpolation is the result of calculating data points within a range of discrete set of known data points, in our case the range of the camera’s position at both start and end snapshots.

The calculation in between two snapshots generates a set of frames that can be used to create a video. Figure 4.16 shows how this mechanism works. The mechanism defines a range of points in-between Snapshot #1 and Snapshot #2 camera’s positions. With the position and focus point, it is trivial to determine the camera’s direction in the 3D space. Having defined the duration of the transition in-between snapshots, our system determines the camera position and direction for each intermediate
frame. For example, by default, the system created 5 second videos with 30 Frames per Second (FPS) for two snapshots. This way the system had to generate 150 frames. The interpolation mechanism had to calculate the camera position and direction for the other 148 frames. Our first solution consisted in calculating the camera position for each frame with a fixed direction on the last snapshots’ focus point (the Snapshot #2 focus point). With the help of a JS library, Tween.js\textsuperscript{11}, the computation of the positions was easy to implement. This library provides a tweening engine that helps the creation of smooth animations. The frames were captured with a JS method that converts the HTML canvas’ content to an image (HTMLCanvasElement.toDataURL\textsuperscript{12}).

During this stage, our solution also computed the necessary data requests that had to be made, in order to present the particles while the camera trajectory was being computed. This was possible because each snapshot taken had the information of its timestep (and the visualization date). The interpolation mechanism is the same applied to the trajectory, but to determine in which intermediate frames the particles will be loaded. By knowing the date corresponding to the first and last snapshots, it was trivial to load the data when interpolating camera positions. For example, if the first snapshot corresponded to the 1st of July and the second snapshot to the 5th of July, the intermediate frames would have to represent linearly the 2nd, the 3rd and the 4th of July.

All the generated frames were stored, to be assembled later together to make a video. For the process of assembly, we used FFmpeg\textsuperscript{13} program to assemble all the frames and export it as a video.

At this point, we were still researching for the best solution for our software. Our final goal was to have a feature that allowed the users to export videos from the software. The video that was presented to the IMAR/MARE team was a 5 second fly-through video that showed the plume dispersion in the deep-sea. The presented video demonstrated the camera performing a trajectory and particles represented with color coding, as seen on Figure 4.17.

### 4.4.6 Improvements

The greater achievement with this first prototype was the interaction with a 3D scene, loading and visualizing data, an interface and the mechanism to generate the necessary frames to create animated

\textsuperscript{11}https://github.com/tweenjs/tween.js/
\textsuperscript{12}https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement/toDataURL
\textsuperscript{13}https://ffmpeg.org/
videos. Despite all the achievements, there was clear room for improvement with this first solution.

When generating the intermediate frames between two snapshots, the system had to compute each frame, as seen before. We found out that there were discrepancies between what was expected and what the application generated. We noticed that this could possibly be caused by the JS asynchronous nature. When loading data, the answer is not immediate, causing the system to behave differently for the same parameters every time we generated the frames. At this point, the system was still using the first data loading mechanism, because the data was structured in binary files. For each request, the frontend application had to translate it into JS objects. The time to process these files would increase with data size, so this was not a feasible process. This acknowledgement led us to develop the data preprocessing script (Section 4.2) and a REST API (Section 4.3).

As mentioned before, when generating the camera trajectory in between two snapshots, the program would get the second snapshot’s direction and apply it to all intermediate frames. This mechanism made the camera perform a rotation motion. An improvement for this would be to develop the possibility for the camera to perform a different motion, like a translation motion.

We also believed that the quality of the video could improve significantly in terms of video quality, color, texture, information presented on the video display and visualization of the particles’ flow. We also noticed that it was necessary to develop methods to allow users to change more settings regarding video quality, such as frame rate or definition.

Besides the exporting feature, we also understood the interface had to change, either by adding more features or by organizing it in a different way. We decided to talk to the experts and receive their feedback, to later iterate through it and apply their suggestions.
4.4.7 Feedback

After the development of the first prototype, informal evaluations were performed by two experts in this field of study, one from the IMAR/MARE team and one from MARETEC, to validate the interface and the animation created. We performed an informal evaluation, to identify necessary changes and missing features.

During the informal interviews, the participants indicated that a user-friendly and intuitive interface were two important aspects missing in similar applications. The ability to generate animations with almost no effort or learning skills on a new tool seemed certainly attractive.

Listing the exported videos for each session would be a compelling feature. This would allow them to generate multiple videos for the same list of snapshots with possibly different parameters of visualization. This way any comparison between videos would be easier than in any other way.

They proposed the implementation of more interaction features for the camera roll, such as supporting the order rearrangement of snapshots (e.g. place the last snapshot on the first place), selecting time duration for each transition (by default was 5 seconds) and picking a non-linear camera motion.

The interface did not provide any way to change the particles’ visualization properties. By default, our application only allowed to visualize the particles’ concentration with a default color scale. Providing a way to change the visualization property and visualization scale were two of the suggestions from the experts.

Besides all of the described suggestions, the participants mentioned that it would be useful to have more control over the visualization and video rendering options. For example, choosing between 30 or 60 FPS, or applying different color or textures to the 3D objects (deep-sea terrain or background scene). It was also mentioned that it would be important to have an overlay legend displayed on the video, telling what was being visualized (e.g., the property, visualization scale, color scale).

In conclusion, the participants’ feedback was really important to understand that the direction we took with this first prototype met their needs that further development with more educated decisions would build a tool that facilitates their daily work.

4.5 Sedify: Operational Prototype

After the demonstration of the first prototype (section 4.4) and the informal review by the experts we developed an operational prototype based on their comments and suggestions. The development of this prototype took into consideration the work presented in the Related Work section. We extended the features presented in the first prototype, applied the changes suggested by the experts and made corrections to the system’s performance.

In this section, we will explore not only the improvements made on the first prototype, but also a feature-oriented explanation of the operational prototype. At this stage of development, the data preprocessing (Section 4.2) and REST API (Section 4.3) were already integrated with the client application. Figure 4.18 shows a main screen example of Sedify.
4.5.1 Improved Interface and Data Loading

As seen before, the interface developed during the first prototype stage lacked some features. To overcome this lack of features and improve the interface, we considered implementing the Timeline Bar, similar to the one described in Section 4.4.2. This way the interface was divided by the Timeline Bar (top bar) and a multiple purpose bar (bottom bar).

The Timeline Bar (Fig. 4.21(a)) was used to place a slider to make it possible for the users to navigate through time. The ability to freely explore the data spatially over time is a key feature of this software. The free spatio-temporal exploration ability of our software is possible due to the operations of requesting the data from the server to render it in the frontend, and the mechanisms to interact with it in 3D. As seen in the Section 4.3, it is possible to retrieve all the data for a specific date with the use of the timestep, since all the visualization aspects and data are refer to the current loaded timestep.

The multiple purpose bar (Fig. 4.19) includes features to capture snapshots of the visualized timestep, edit and organize the captured snapshots to make a video on the Camera roll, navigate in time, export a video, watch all the exported videos and edit the visualization appearance of the scene and/or video. This bar also provides a color scale that indicates the current visualization scale, property of visualization and the color gradient chosen.

![Figure 4.19: Multiple purpose bar. The Camera roll menu being presented.](image)

The color scale with the current selected particles' property and visualization scale was placed in the
center of bottom bar’s header. This element is always visible and updated when the property, visualization scale or color scale are changed. This element was carefully put in this part of the UI to be easily visible. An example can be seen on Figure 4.20.

![Color scale and visualization options.](image)

**Figure 4.20:** Color scale and visualization options.

## Time Interaction

The application's visualization state is coordinated by the global variable, *timestep*. The Timeline Bar (Fig. 4.21(a)) allows to interact with the date of visualization by dragging the cursor to the left and right of the slider and changing the timestep. It is also possible to change the timestep by clicking the buttons (Fig. 4.21(b)) to increment or decrement the current timestep.

The Timeline Bar was located on top of screen and was always visible, providing a visual cue of the current date of visualization and an easy way to select any timestep of the current visualized dataset. This element is composed by a text indicator of the current date of visualization, two text indicators that show the begin and end date of the loaded dataset, and the slider. Four buttons (Fig. 4.21(b)) were also added to the interface to move forward or backwards in time, by incrementing or decrementing the timestep. These buttons allow to increment or decrement the timestep by one or 5 units, by default.

Once a new timestep is chosen, the data that corresponds to the timestep is requested to the server through the REST API (Section 4.3). The endpoint used is `/api/sediments/{filename}/{timestep}`, being the timestep, the one selected using the Timeline or the buttons. Once the API response (JSON) is received, the particles are updated in the 3D scene, and all the previously loaded data is cleared. The text indicator of the current visualized date is also updated upon receiving this response.

![Timeline Bar](image)

**Figure 4.21:** The interface elements used to interact with the time component of the application.

## 3D navigation

In order to allow the user to freely navigate in the 3D scene, it was crucial to develop controls to change the viewpoint and the viewing direction of the scene. As shown before, common controls used for this are pan, zoom and rotate controls. The mouse can trigger any of these controls, since each one of its buttons is mapped to a different control. The Right Mouse Button (RMB) is responsible for the rotation control, the Middle Mouse Button (MMB) is responsible for the zoom control and the Left Mouse Button
(LMB) for the pan control. These mouse can be easily programmed to behave differently, or to change the buttons’ functions.

We opted to implement orbit rotation controls, meaning that the view rotates around a point of interest. The point of interest is changed when the user chooses to pan. We also implemented a hotkey to change the point of interest with more precision by performing Alt+LMB in the requested point. Pressing and dragging the LMB, allows the user to rotate the model by changing the viewing angle. We opted to choose the turntable rotation system for this, where left and right rotate the camera around the viewport’s global Z axis.

The zooming control moves the camera backwards and forwards. It is possible to zoom by holding the MMB and move the mouse up or down, zooming in or out, accordingly. The panning control moves the view up, down, left and right by holding and dragging the RMB.

**Flow Visualization**

As discussed earlier, pathlines are commonly chosen to visualize drift trajectories [OBRV15, BW11]. A pathline describes the trajectory of a particle over time. Due to our data and particles’ structure, it seemed an obvious choice to choose this method to visualize the flow field.

For the flow visualization to be effective, it has to focus on the representation of the field’s magnitude (speed) and direction. A pathline segment that represents a travelling particle is called pathlet. A pathlet can convert a particle’s position into a pointer of speed and direction. Pathlets are represented as line segments that start a the current position of the particle and connect each new position of the particle to its previous position until reaching a location past a threshold (e.g. tail size).

![Tail size: 10](image)

Figure 4.22: Comparison of three moving particles with a tail size of 10 units. A: Represents the current state of the particle. B: Represent all the ten past states.

The user has the ability to choose the tail size, in other words, the number of a particle’s past positions. This option gives more or less visual weight, with more or less historical trajectory data. The current position is more opaque than past positions, providing a perceptual hint of the pathlet's
direction, even in static screenshots, as seen in Figure 4.22. As particles move faster, their pathlets trace out a longer path. It is possible to conclude that pathlet P1 (Fig. 4.22) was the fastest and P2 the slowest of the three particles group (P1, P2 and P3).

Our representation method effectively shows the speed and direction of moving particles. Figure 4.23 shows a more complex visualization, in which more particles are represented. The pathlets have a tail size of 5 units. The three screenshots show 10 days of simulation, the first one is ten days apart from the last one.

![Figure 4.23: 10 days of simulation of Patorra area, between the 16th of January and the 26th of January of 2011. Flow visualization example with a 5 unit tail size.](image)

**Color coding**

As said before, color is a commonly chosen visual channel, used mostly as a scale. This visual channel works well with flow visualization methods. Color adapts very easily to different representations and is applied in many different contexts [WKP14]. The color scheme plays a big role in the effectiveness of the visualization. We allow users to select and play with different color schemes. We used library Chroma.js, because it provides easy access to color conversion and color scales (Fig. 4.24).

This modifier assigns colors to particles based on selected properties. The UI offers input fields to allow the user to select between a two-color gradient or default color gradients (e.g. rainbow color gradient). Our system assigns colors to particles based on one of their properties, providing a simple, yet effective method to visualize scalar per-particle quantities.

### 4.5.2 Creating and Exporting Videos

Our application provides the ability to take snapshots of the 3D scene, in other words, the ability to keep a record of the snapshots taken at given timesteps. The user can combine this mechanism with the navigation controls, explained earlier and build a robust compilation of pictures of the 3D scene. As already stated, this application automatically iterates through this collection and exports it as a video.

As seen in the Section 4.4.4, a snapshot is a composition of the camera position and focus point in the 3D scene, associated to the timestep when it was taken. The user can go back and forward in time, and take as many snapshots as wanted, by clicking the button "Snap" presented in the multiple purpose bar (Fig. 4.19). When this is done, a new snapshot gets added to the last position of the Camera roll. This new entry shows an image of scene that was captured, and at the bottom left, provides the information of the duration of the video up to that frame. This is shown on Figure 4.25.
Interact with the Camera roll

During the informal evaluation, the participants considered the concept of taking snapshots intuitive, but lacking features. Some of the features mentioned were selecting the time duration of a transition between two snapshots, the ability to rearrange the order of snapshots in the Camera roll and camera non-linear motion. All of these features were implemented during this stage of development.

Besides the features mentioned, we decided to implement other necessary features. We included a load feature, that allows users to load a particular snapshot. This will load the corresponding data from the snapshot's timestep (particles and date) and will place the camera at its position and direction. This feature may be useful when a user wants to recreate a snapshot or just copy and duplicate the snapshot. We also included features to delete a particular snapshot or all of them, clearing all the
collected snapshots. We have placed the “Clear all” button always visible, for easy access, since this may be a recurrent situation.

Figure 4.26: Snapshot screen. Is is shown the date, a picture of the scene, transition properties, delete and load buttons.

When the user hovers the cursor over a particular snapshot in the Camera roll, it gets larger, providing a visual cue that each one of these elements is clickable. Once the user clicks a snapshot, a new display is opened (Fig. 4.26). The date is presented at the top of the display. Right below the date, a picture of the scene is shown, providing a visual cue of the camera’s position and direction. Three dropdown lists are presented: the Easing, Duration and Movement Type. The Easing list allows the user to select linear and non-linear camera motion. This applies a different rate at which the camera position changes between two snapshots when rendering a video. The Duration provides a list for the user to select the duration of the interpolation between two snapshots, in seconds. The Movement Type list allows the user to select between Rotation and a Translation camera motion, between each two snapshots. All these properties are applied to the transition between the selected snapshot (the one that was clicked) and the next one on the list.

It was also mentioned that a relevant feature to include would be the possibility to rearrange the order of the snapshots’ list. A feature was implemented that allows the user to drag a snapshot and drop it anywhere inside the snapshots’ list. The Figure 4.27 shows a snapshot getting dragged to a different position. When a snapshot is getting dragged across the list, all the snapshots start shaking, providing a cue that they can be moved away from their positions. The video generation takes into consideration the final order of the snapshots.
Rendering a Video

Once the user decides to generate a video with the snapshots taken and edited parameters, the interface offers a button “Render” that, when clicked, opens a new screen that displays the duration of the video to be exported and its framerate (Fig. 4.28). This display provides a confirmation button to start rendering the video. Once clicked, the display is updated and shows a loading bar, that represents the progress of the exporting process, and two buttons “Stop&Save” and “Cancel”. The first button will stop the rendering process, and save the processed video until that time. The second button simply stops the process and discards everything.

Although part of this process has been explained before in this document (Section 4.4.5), it was improved during the development of the operational prototype. As explained earlier, the video generation consists on generating frames that make an animation when assembled together. The frames generated
consist on the transition frames between two snapshots of the list of snapshots taken (Fig. 4.16).

The asynchronous nature of the first prototype rendering mechanism was identified as one of the issues that had to be improved. Our developments made the asynchronous calls wait for the previous requests’ answers. This made it possible to request large chunks of data, while producing smooth animations. We used CCapture\textsuperscript{14}, a JS library that facilitates capturing from an HTML5 canvas animations at a fixed rate. The camera motion was calculated with the help of Tween.js, as seen before. The videos are exported in the WebM\textsuperscript{15} file format. Figure 4.29 represents the algorithm implemented to calculate the intermediate frames for in-betweening.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{rendering-mechanism_diagram.png}
\caption{Rendering mechanism. Fluxogram for intermediate frames computation.}
\end{figure}

\textsuperscript{14}https://github.com/spite/ccapture.js/
\textsuperscript{15}https://www.webmproject.org/
information that is stored on the first snapshot of each snapshot pair. This information contains the camera position and direction, the motion type (Fig. 4.30), the duration of the transition, the current snapshot's timestep and the next snapshot's timestep, as well as the easing function (Fig. 4.31). These are the necessary parameters to calculate all the frames for each transition. The user is able to select between 30 or 60 FPS, the latter will produce twice the amount of frames, but a smoother transition.

The rendering mechanism supports two different camera motion types: the rotation and the translation (Fig. 4.30). When generating the frames for the rotation motion, for every new camera position, the camera direction is pointing towards the last frame's direction point. As seen in the Figure 4.30, for every new generated frame the camera is always looking at a fixed point. On the other hand, the translation motion is created by applying the same direction and calculating a new point to look at, for every new frame.

To make the camera motion non-linear, the user can select a non-linear easing function. The system provides a good amount of options, as seen in the Figure 4.31. The selected function calculates the camera position for all intermediate frames. A linear function will, simplistically, move with the same speed over time, whereas on a non-linear function, the speed will vary over time. There are three types of non-linear functions: Ease In, Ease Out and Ease In Out. With a Ease In function, the motion will start slowly and accelerate over time, the opposite goes for Ease Out. For the Ease In Out, the animation starts and ends slowly, accelerating and reaching its peak speed in the middle.

As mentioned by the experts during the interview of the first prototype, a video has to display the visualization related information to be informative, otherwise it would be meaningless. We have developed a way to display a legend of the visualization (screen overlay), that shows the color scale associated with the particles’ property of visualization and the scale, and the date of visualization. The legend can be placed on any of the corners of the screen. An example of this is shown on the Figure 4.32(b).
After creating and exporting a video, the UI offers a menu where it is listed. The UI provides an always visible button "Videos list", on top of the bottom bar. Once clicked, it opens up a display with all the videos exported during the current session. The user can click on any video to load it, play it and/or download it. This menu provides the option to watch the video in the small video player or in full screen.

4.5.3 Change options

The Options menu is divided into three sub-menus: Particles, Geometry and Scene and Video Render. The three menus are shown on Figure 4.33.

The Particles sub-menu allows to tweak particles’ visualization parameters. It is possible to choose the particle size, toggle particle tail, to choose the particle tail, select the visualization property (variable), choose a default color gradient (e.g. rainbow color scale) or to make one out of two colors, and select the visualization scale. This screen is shown on Figure 4.33(a).

The Geometry and Scene sub-menu allows to tweak parameters related to the deep-sea terrain and the scene in general. It is possible to choose a color for the deep-sea terrain, select a color for the background, select the opacity of the deep-sea terrain, toggle the deep-sea terrain wireframe and show/hide it. This screen can be consulted on Figure 4.33(b).

The Video Render sub-menu serves essentially to apply a different framerate, to choose at which corner of the video to place the legend or to show or hide the coordinate labels. The latter can be used to give geographic context. This screen is shown on Figure 4.33(c).
List of rendered videos

Here you can select and watch previously rendered videos.

- Video 0
- Video 1
- Video 2
- Video 3
- Video 4

Figure 4.32: The users are able to watch and download all the created videos during the session.
Figure 4.33: The Options’ sub-menus.

(a) Particles sub-menu.

(b) Geometry and Scene sub-menu.

(c) Video Render sub-menu.
Chapter 5

Evaluation

In this chapter we present the methodology chosen to evaluate our work. We took into consideration both experts and general users (engineering students or with scientific background) for our analysis. We also present a comprehensive set of results obtained from the questionnaires and its analysis, as well as all the feedback collected during the tests and the interviews with experts and its discussion.

It is important to decide the metrics used for evaluation, since they will determine if this tool successfully meets our goals. As a tool, we wanted to create an intuitive and user-friendly interface that would be easy to use to produce quality visualizations.

Our evaluation was done in two separate ways: by interviewing experts in this field of study and via usability tests. We believed that both methods would reveal whether this tool solves the proposed problem and what can be done in the future to achieve better results.

The experts were expected to give us feedback on the operational prototype as well as videos generated by the application. They suggested improvements based on their experience and verified if the animations met the commonly used standard. General users tested this system by executing tasks and answering a questionnaire. This testing stage provided information on the overall experience and usability of our system.

For all the tests and presentations made during the evaluation stages, a dataset referring to the Patorra area was used. We chose the largest dataset IMAR/MARE had produced with 30 GB of total size.

5.1 Experts’ Feedback

We believe that to successfully evaluate this tool we had to have conversations with experts of the oceans’ field and researchers that use tools to analyze ocean phenomena. We conducted all of our tests at this stage, with experts from MARETEC and IMAR/MARE.

5.1.1 Methodology

Each evaluation session of Sedify has comprised 3 main stages:
1. Introduction

2. Interface and videos evaluation

3. Questionnaire

**Introduction**

We started our introduction by greeting the participants and by explaining that the main goal of this work was to assess the quality of the animations produced and if the developed tool suited their needs as professionals. We explained the implemented features, the use case and the data they were visualizing.

**Interface and videos evaluation**

In the experimentation stage, the experts had the opportunity to explore the system, create animations and watch already created videos. We wanted to assess if the interface and the animations met or exceeded their standards.

We conducted a demonstration via video-call for the experts that could not be present. Since we were not evaluating the usability or experience of the tool at this stage, this did not consist in a problem. For those who could be present, they were allowed to explore the system by interacting with it. Either way, the experts were explained the process to create an animation and visualize data. The system and the implemented features were explained in detail. Previously created animations were shown and the experts gave their opinions on those.

**Questionnaire**

At the end of the evaluation phase, each expert was required to answer a set of profiling questions, like their role and expertise. The questionnaire also contained 13 sentences that the professionals were expected to evaluate with a 7-point Likert scale (7 meaning that they totally agreed). These sentences were mostly about their point-of-view on the system and visualized animations. The sentences were:

1. I consider Sedify useful in my work.
2. Sedify would make my work easier.
3. I believe that Sedify is a system easy to work with.
4. I think I would learn easily how to use Sedify.
5. At my workplace, I have the necessary conditions to use Sedify.
6. At my workplace, I have the opportunity to use Sedify.
7. I have enough knowledge to use a system like Sedify.
8. It would be necessary to use Sedify almost every day.
9. Sedify has the necessary tools to perform data analysis.
10. I trust the Sedify’s results.
11. I believe Sedify works well and without problems.
12. A system like Sedify is the future of 3D data visualization.
13. Given the opportunity, I would prefer to use Sedify instead of traditional methods.

After evaluating these sentences, the experts’ had to answer the three following questions:

– What did you like the most about this software?
– What did you like the least about this software?
– What limitation do you think this system has?

Given that a system with good usability does not necessarily mean it is useful for its end-users, we want to assess the usefulness and utility with these sentences and questions.

5.1.2 Results

In this section, we will present and discuss the results obtained in this evaluation stage.

Participants’ profile

This evaluation stage was conducted with 5 participants, 3 were male and 2 female. Three participants were inquired in person and 2 were inquired via video-call. For the purpose of distinguishing between participants, they are labeled as E1 (Expert 1), E2 (Expert 2), and so on. The participants were researchers from MARETEC and IMAR/MARE. E1 is an Associate Professor at Universidade dos Açores and, E2 is an Associate Professor at Instituto Superior Técnico, Universidade de Lisboa. The participants were experienced with data visualization software of ocean data and/or visualization of flow. Two of the participants work in this field of research, and are experienced in visualizing and analyzing sediment dispersion data (E1 and E4).

Questionnaire results

The goal of the presented questionnaire was to assess if the choices we made with our application were able to provide the experts with the necessary features to visualize data and a user-friendly interface. The questionnaire revealed useful information on the state of the software and the features the experts need the most. The results of the questionnaires are presented in the Table 5.1.

It was consensual that the experts considered the developed system useful in their work (IQR=0). This sentence had a minimum of 4, given by E3, since this expert’s work is not related to the analysis of sediment dispersion in the deep-sea. Sedify was believed to make their work easier (IQR=1), except once again, for E3. They believed that this system was easy to work with (IQR=1) and that it was easy...
Table 5.1: Experts questionnaire with median and IQR calculated from the responses. Minimum and maximum values also presented.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Median</th>
<th>IQR</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - I consider Sedify useful in my work.</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>2 - Sedify would make my work easier.</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>3 - I believe that Sedify is a system easy to work with.</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>4 - I think I would learn easily how to use Sedify.</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>5 - At my workplace, I have the necessary conditions to use Sedify.</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6 - At my workplace, I have the opportunity to use Sedify.</td>
<td>7</td>
<td>0</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7 - I have enough knowledge to use a system like Sedify.</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8 - It would be necessary to use Sedify almost every day.</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>9 - Sedify has all the necessary tools to perform data analysis.</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>10 - I trust the Sedify’s results.</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>11 - I believe Sedify works well and without problems.</td>
<td>7</td>
<td>0</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>12 - A system like Sedify is the future of 3D data visualization.</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>13 - Given the opportunity, I would prefer to use Sedify instead of traditional methods. (IQR=1).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.1: Distribution of the experts’ answers plotted as a boxplot diagram.

to learn how to use it (IQR=1). Their workplace offers both conditions and opportunity to use this system (IQR=0). The results seem trustworthy and the system seems to work well with no problems (IQR=0). They believed that a system like this seems to be the future of 3D data visualization and prefer this over traditional methods (IQR=1).

Sedify does not provide all the necessary tools to perform data analysis (IQR=0). We were expecting these results for this particular sentence, since this was never the intended purpose of this tool. We implemented basic data analysis features, to allow users to quickly and effortlessly create animations.

This tool may not be necessary every single day (IQR=2). We have learned that experts need these tools, mostly when it is relevant to show or present results in a conference or in political forums, for example. E1 and E4, out of the 5, were the participants who believed they would benefit the most from this tool.

The questionnaire had three open questions that we devised to help us understand how a system like ours would affect, positively or negatively the experts’ work. The questions consisted on understanding what the experts liked, disliked and believed was a limitation. Some of the answers given to what they liked the most about this software:
- "Making videos very easily with a visualization that looks great." (E3)

- "User-friendly and intuitive interface. Applying the concept of existing tools, but with better user experience. The system seems faster than similar tools." (E5)

- "Making videos to be used on presentations and data-analysis. The simplicity of the system, being very easy to use." (E1)

- "Being an interactive 3D system. Big potential to be used with Virtual Reality (VR)." (E2)

- "Nowadays, having these systems on the web is fundamental. The system has a simple and enjoyable interface. The control over the color and the particle size is fundamental and not always available in similar systems." (E4)

The experts have not disliked many things about this software. Some of the answers given to what they disliked the most about this software:

- "It is not an alternative to systems that offer more complex visualization features." (E4)

- "Some menu names may be confusing, since there is not a standard for these applications. I thought the Render menu served to render the scene in 3D." (E1)

- "Personally, I don’t think the date slider is straightforward to choose a date." (E3)

Besides that, they have pointed out some limitations they found. Some were:

- "I would like to watch a video before rendering it." (E4)

- "I believe this system needs to store sessions, so I don’t have to configure each time I start the application." (E5)

- "It needs to show the area of visualization in a world map, giving a geographic perspective." (E3)

- "I would like to export videos in different formats." (E1)

**Discussion**

It is possible to say that we have achieved our goal of creating a user-friendly interface. That was mentioned more than once by the experts. We also believe that we have achieved our goal of building a useful system with a better user experience than similar tools. The experts appreciated the particle system and the visualization options provided by this system. It was interesting to see VR mentioned as a potential future development for this too, by one of the experts.

Our system does not provide complex visualization features that other systems do. This tool was developed to be a web application to provide the necessary visualization options and 3D interactivity to perform the activity of building an animation. It is clear that this tool it is not an alternative to other more complete tools. However, the experts also mentioned that more complete tools have a less user-friendly
interface and interaction, associated with a steeper learning curve. This way our tool seems to be a good alternative.

One expert (E1) mentioned that some menu names like “Render” may be confusing. We have followed the common convention for these systems. It may be possible that this means something completely different in different environments, but that is out of the scope of this work.

E3 mentioned that the date does not seem straightforward to choose on the timeslider. To check this, we planned usability tests, to better understand if not only this particular element, but if this system is usable. The tests and results are presented in the Section 5.2.

The limitations the experts pointed out can be seen as suggestions for the future of this tool, or other similar tools. We believe that no big limitation was pointed out. Watching a video before rendering it, may be possible with preprocessing rendering mechanism that allows the visualization of an animation already processed once requested. The developed REST API can be adapted to allow the storage of the session. The browsers built-in local storage are also an option to the expert considered limitation.

In conclusion, we have received really important feedback from the experts that not only validated the work done, but also provided quality suggestions for future development. Sedify can be considered a new and better way of creating sediment plume dispersion videos. End-users can easily and quickly create videos with this tool. This is meaningful due to the fact that the experts have knowledge in the matter. We can conclude that Sedify is a useful tool.

5.2 Usability Tests

User testing is the most used empirical method for evaluating interfaces [Nie94]. Usability defines the ease of use and learnability of a system. In order to assess this, we defined a set of 3 tasks and evaluation metrics, to understand how well users performed. We decided on tracking the success rate for each task, the finishing time and the users'subjective satisfaction.

This evaluation stage was conducted with 20 participants and took place in private houses and university rooms. All the tests were carried out under equal conditions for all the participants to minimize behavior deviations.

5.2.1 Methodology

Each evaluation of the usability evaluation has comprised 4 main stages:

1. Introduction

2. Free experimentation

3. Task execution

4. Questionnaire
Introduction

We started our introduction by greeting the participants and explained that the main goal of this work was to assess the usability of the application. We explained the implemented features, the use case and the data they were visualizing.

We explained how the software was supposed to work and that the main goal of this evaluation was to determine if it is easy to generate animations and visualize spatiotemporal data.

Free experimentation

The participants were given three to five minutes to explore the application without any instruction and get familiarized with it (Fig. 5.2). We also took a few minutes to answer any questions regarding the interface and the use case.

![Figure 5.2: User experimenting the application.](image)

Task execution

After exploring the system, the participants were required to execute tasks. The tasks’ order was decided to assess if the users learned how to use Sedify by making less errors as they were performing the tasks.

Each task was broken down into a subset of actions the participants must do. If they performed this procedure differently than it was expected or do not perform at all, that was considered a fail (miss). Otherwise, it was considered to be right. One task could be completed with as many errors as the number of the subset’s actions. These actions served to better evaluate the tasks’ performance, and provide an insight if the application provides the right mental model. It is important to stress that these actions did not have to be performed by any particular order, nor were explicitly asked to be performed. We tracked how much time the users took to finish each task with a chronometer.

For Task 1, we prepared a session with 5 snapshots (Fig. 5.3). The first snapshot was taken at 10/01/2011 at 0:00, the last snapshot 11/01/2011 at 06:00, and the total duration of the video to be exported was 40 seconds. The participants were asked to let us know when the video starts, ends
an how long it takes. We believed that this task would allow us to determine if a user finds intuitive the interaction with a session prepared to export a video. In order to do this, the user had to click on the first snapshot to check for what date it was taken and the last snapshot for the last date (begin and end date). The total duration of the video corresponds to the time that appears on the bottom left corner of the last snapshot. The task stopped when the user answered the three questions.

With Task 1, we wanted to discover if the users had difficulty in understanding and interacting with the snapshots. For this task, the participant has to discover begin date, end date and the duration of the video to be exported. We defined the following actions:

- **1-a:** The participant gets the begin date right This was achieved by opening the first snapshot and checking the date on the header.

- **1-b:** The participant gets the end date right This was achieved by opening the last snapshot and checking the date on the header.

- **1-c:** The participant gets the total duration right This was achieved by looking at the bottom-corner of the last snapshot on the horizontal list of snapshots.

![Figure 5.3: The session presented during to the participants on Task 1.](image)

For Task 2, we presented an existing video and asked each participant to recreate the most similar video possible (Fig. 5.4). The video presented started on 10/01/2011 at 0:00 and ended on 28/01/2011 at 0:00, with a duration of 15 seconds. In this video the particles’ trace was activated and the camera movement is a top view descending camera effect (zoom). We wanted to understand if participants could recreate the video. The participants were able to watch the video a few times until they were comfortable to start. The task ended when the user clicked the button to start rendering the video.

With Task 2, we wanted to discover if the users were able to watch a video and recreate it using the system. We believe that this task was the most complex of the three. The user has to understand different concepts by watching the video, to later apply them when interacting with the application. For this task, the participant had to watch an existing video and recreate it. We defined the following actions:

- **2-a:** The participant chooses the right begin date This was achieved by taking a snapshot on the start date of the video.
– 2-b: **The participant chooses the right end date** This was achieved by taking a snapshot on the end date of the video.

– 2-c: **The participant applies the right total duration** This was achieved by changing the duration of the transition on the first snapshot to 15 seconds.

– 2-d: **The participant activates the particles’ flow visualization** This was achieved by opening the Particles sub-menu (Fig. 4.33(a)) and click the check-box “Toggle particle tail”.

– 2-e: **The participant recreates the descending camera effect** This was achieved by taking the first snapshot at a higher altitude than the last one. The interpolation recreated the effect.

– 2-f: **The participant starts the rendering process** This was by opening the render menu and clicking "Render".

![Video presented on Task 2.](image)

For Task 3, the participants were given a screenshot of a session and asked to apply the necessary changes to their session to get a similar snapshot (Fig. 5.5). This session had the Depth variable chosen, a blue background, the deep-sea was grey and the scale of visualization was [900m, 1000m]. We wanted to understand if the participants would be able to recreate a session intuitively by applying different settings. The task ended when the participant believed that it was finished.

With Task 3, we wanted to assess if the users were able to replicate a session by giving them a screenshot of session (Fig. 5.5). For this task, the participants were given a session screenshot and has to recreate it. We defined the following actions:

– 3-a: **The participant chooses the variable Depth** This was achieved by opening the Particles sub-menu (Fig. 4.33(a)) and selecting “Depth” on “Select variable” list.

– 3-b: **The participant applies a different color to the background (blue)** This was achieved by opening the Geometry and Scene sub-menu (Fig. 4.33(b)) and selecting the blue color on “Select the bathymetry color” color picker.

63
- 3-c: **The participant applies a different color to the deep-sea (grey)** This was achieved by opening the Geometry and Scene sub-menu (Fig. 4.33(b)) and selecting the grey color on the "Select the background color" color picker.

- 3-d: **The participant changes the scale to [900m, 1000m]** This was achieved by opening the Particles sub-menu (Fig. 4.33(a)) and writing "900" on the "Select the start value for the selected property", and "1000" on the "Select the end value of the selected property".

![Figure 5.5: Screenshot presented on Task 3.](image)

**Questionnaire**

After finishing the tasks, the participants were asked a questionnaire. This questionnaire contained sentences the participants had to evaluate with a 5-point Likert scale, where 1 means "Strongly Disagree" and 5 "Strongly Agree". Two demographic questions were also included for age and gender.

The questionnaire had sentences to reveal the users’ profile, their experience of using our system and the System Usability Scale (SUS). SUS is subjective metric, that consists of a 10 item questionnaire. Since this system was designed having in mind the experience of manipulating a video editing tool, we wanted to assess if our participants felt comfortable using these sort of systems. The SUS may serve to evaluate our system regarding its usability in terms of the effectiveness, efficiency and satisfaction.

The participants had to evaluate the following sentences:

- **Profile**
  1. I feel comfortable manipulating 3D environments.
  2. I feel comfortable using video editing tools.

- **Experience**
  3. It was easy to identify the particles’ trajectory.
  4. It was easy to pick a new date on the Timeline bar.
  5. It was easy to determine the video length.
  6. It was easy to change settings and appearance (e.g. scale, variables, colors).
• System Usability Scale

7. I think that I would like to use this system frequently.
8. I found the system unnecessarily complex.
9. I thought the system was easy to use.
10. I think that I would need the support of a technical person to be able to use this system.
11. I found the various functions in this system were well integrated.
12. I thought there was too much inconsistency in this system.
13. I would imagine that most people would learn to use this system very quickly.
14. I found the system very cumbersome to use.
15. I felt very confident using the system.
16. I needed to learn a lot of things before I could get going with this system.

After a user gives his responses the SUS scoring is obtained by:

1. Convert the 1-5 scale to 0-4 scale (being 4 the most positive answer).
   
   For odd-numbered items: Subtract one from the answer.
   For even-numbered items: Subtract the answer from 5.

2. Add up the converted responses, producing a value between 0 and 40.

3. Multiply the value by 2.5, turning it into a value between 0 and 100.

Despite the SUS conversion producing a value in a range of 0-100, these values are not percentages, being the average score 68.

5.2.2 Results

In this section, we will present and discuss the results obtained in this evaluation stage.

Participants’ profile

This stage of evaluation was conducted with 20 participants, 12 were male and 8 were female, 12 with ages between 18 and 24, 5 between 25 and 36 and 3 between 35 and 50 years old.

The participants were given three sentences and asked to evaluate them with a 5-point Likert scale to better understand their profile. By analyzing the answers given (Table 5.2), it is possible to conclude the the participants felt comfortable manipulating 3D environments (IQR=1) and felt comfortable using video editing tools (IQR=1).
Table 5.2: Profiling questions with median and IQR calculated from the responses.

<table>
<thead>
<tr>
<th>ID</th>
<th>S. Rate</th>
<th>Time (s)</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>S. Rate</th>
<th>Time (s)</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>S. Rate</th>
<th>Time (s)</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>57</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>47</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>30</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>80</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.75</td>
<td>75</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>50</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.67</td>
<td>120</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>0.5</td>
<td>150</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>64</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.67</td>
<td>57</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>1</td>
<td>62</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>80</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>125</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>0.67</td>
<td>14</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>0.67</td>
<td>291</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>1</td>
<td>48</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.67</td>
<td>129</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>1</td>
<td>24</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>20</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>150</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>1</td>
<td>54</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>35</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>192</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>35</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>15</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>370</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>75</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>20</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>214</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>0.75</td>
<td>161</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>31</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>263</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>1</td>
<td>105</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>0.67</td>
<td>15</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>1</td>
<td>353</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>203</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>26</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>273</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>178</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>0.67</td>
<td>37</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>1</td>
<td>175</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>162</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>0.67</td>
<td>45</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>1</td>
<td>136</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.75</td>
<td>90</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>32</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>314</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>180</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>25</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>153</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.75</td>
<td>161</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>65</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.83</td>
<td>241</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>0.75</td>
<td>192</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>20</td>
<td>0.33</td>
<td>35</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>0.83</td>
<td>150</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>1</td>
<td>90</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.9</td>
<td>30.1</td>
<td></td>
<td>0.88</td>
<td>189.9</td>
<td>0.91</td>
<td>110.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Time and Success Rate for each task.

Task execution

At this stage, we kept track of the time and the actions the participants missed while executing each task. The results can be consulted on Table 5.3.

During the execution of the first task (Task 1), one participant failed to get the begin and end dates and four failed to get the total duration of the video to be exported, providing an average success rate of 90% (standard deviation = 19%). On average, the participants took 30.1 seconds (standard deviation = 16.1) to complete the task (Fig. 5.6(a)). We are 95% confident that the mean time it takes to execute the first task is between 23.1 seconds and 37.2 seconds.

(a) Distribution of the time to complete the first task, plotted as a boxplot diagram. (b) Distribution of the first task’s success rate, plotted as a boxplot diagram.

Figure 5.6: Task 1 boxplots.
For the second task (Task 2), one participant failed choosing the begin date, another failed choosing the end date, five failed at creating a 15 seconds video (total duration), six failed at applying the particles’ flow visualization mode and two failed at recreating the descending camera effect. All the participants were able to start the video rendering process. This task has an average success rate of 88% (standard deviation = 13%). On average, the participants took 189.9 seconds (standard deviation = 96.2) to complete the task. We are 95% confident that the mean time it takes to execute the second task is between 147.7 seconds and 232.1 seconds.

(a) Distribution of the time to complete the second task, plotted as a boxplot diagram.  
(b) Distribution of the second task’s success rate, plotted as a boxplot diagram.

Figure 5.7: Task 2 boxplots.

For the third task (Task 3), three participants failed when choosing the Depth variable, one when applying the background color, one when applying the deep-sea color and two when selecting a different scale. This task has an average success rate of 91% (standard deviation = 14.6%). On average, the participants took 110.9 seconds (standard deviation = 58.1) to complete the task. We are 95% confident that the mean time it takes to execute the third task is between 85.4 seconds and 136.4 seconds.

(a) Distribution of the time to complete the third task, plotted as a boxplot diagram.  
(b) Distribution of the third task’s success rate, plotted as a boxplot diagram.

Figure 5.8: Task 3 boxplots.

We have observed no difficulty determining the begin and end date of the video to be exported, during the execution of Task 1. In fact, one participant failed in this test. The participant who failed getting both dates right answered with the begin and end date of the loaded data, the dates that are shown on both sides of the Timeline bar. Surprisingly, 4 participants failed determining the duration of
the video to be exported.

During the second task, the users had to analyze a provided video and recreate it. We considered this task particularly difficult since the users had to perform all the necessary actions to export a video. We noticed that the users had more difficulty in activating the particles' trace, possibly because this option was not visible on the main screen. We observed that the participants did not have much difficulty in recreating the descending camera effect. 5% of the users had trouble in setting the correct time duration. We observed that they became confused by how they were supposed to interact with the transitions between snapshots.

Task 3 was executed after the second task. Participants had to explore the system and have contact with most features. We expected them to have less friction with this task, and that was our observation, since the participants who had failed at activating the particles' flow, a feature present in a hidden menu, did not fail similar actions at this stage.

There were no statistically significant differences between the success rate means as determined by a One-way ANOVA with a \(\alpha = 0.05\) (F(2,57) = 0.29174, \(p = .748073\)), which is expected due to the interface is generally the same for all tasks. However, an analysis of variance showed significant differences between the execution times determined by One-way ANOVA with a \(\alpha = 0.05\) (F(2,57) = 29.71137, \(p = .00001\)). The Post hoc test Tukey's HSD (Honestly Significant Difference), revealed that the HSD is 49.89. This means that minimum difference between means must be 49.89 for significance. The difference between the means of the execution times for each task, have shown that all the three tasks are statistically different. The difference between the first task and the second task is 159.8, the third and the first is 80.8 and the second and the third is 79. This reveals that there is a bigger difference between the first and the second task, and that the difference between the third and the first task is approximately the same as the second and the third task. This was expected due to the fact that the tasks have a different number of steps and complexity, with the second task the most complex of the three. The first task was considered the easiest one.

By comparing all the tasks (Fig. 5.9), more mistakes were made when applying the total duration and activating the particles' flow visualization, during the second task.

In conclusion, the observed results for task execution may suggest that the options that are hidden in sub-menus, could be somehow placed in the main menu.

**Questionnaire**

The questionnaire had 4 questions related to four different parts of the application. We wanted to understand better the user experience. The analysis made to those questions (Table 5.4 and Fig. 5.10) concluded that the users agreed that it was easy to identify particle trajectories, select date on the time slider, determine video length, and applying changes in the settings and the appearance of the scene (IQR=1). When compared the users had more difficulty in determining video length. This is both confirmed by their answers to question 5 of the questionnaire (Fig. 5.10) and the subtask 1-c on Figure 5.9.

Table 5.5 presents the results obtained in the questions of the System Usability Scale and the Figure 5.11 shows the distribution of the average scores after the conversion, plotted as a boxplot. Our
application scored 77.88, which is considered above average, and correlates with a score of "Good". When analyzing the SUS score it is possible to see that the two of lowest scoring answers were "I think that I would like to use this system frequently" and "I needed to learn a lot of things before I could get going with this system", which may indicate that the participants are not the end-users of this system, and had to understand the domain problem to use the system. On the other hand, three of the best scoring answers were "I found the various functions in this system were well integrated" and "I found the
<table>
<thead>
<tr>
<th>User</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>Sum</th>
<th>Final Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>26</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>31</td>
<td>77.5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>25</td>
<td>62.5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>35</td>
<td>87.5</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>32</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>33</td>
<td>82.5</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>29</td>
<td>72.5</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>27</td>
<td>67.5</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>31</td>
<td>77.5</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>33</td>
<td>82.5</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>27</td>
<td>67.5</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>30</td>
<td>75</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>27</td>
<td>67.5</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>29</td>
<td>72.5</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>Average</td>
<td>2.25</td>
<td>3.35</td>
<td>3.2</td>
<td>2.8</td>
<td>3.55</td>
<td>3.5</td>
<td>3</td>
<td>3.65</td>
<td>3.35</td>
<td>2.5</td>
<td>31.15</td>
<td><strong>77.88</strong></td>
</tr>
</tbody>
</table>

Table 5.5: SUS Questionnaire

The system very cumbersome to use”, which indicate that the various functions of the system seemed well integrated and agile to use. Overall, the results are good and show that this system is usable with a **77.88** score.

![System Usability Scale](image)

Figure 5.11: Distribution of the average SUS scores plotted as a boxplot diagram.
Chapter 6

Conclusions

In order to conclude, it is important to revisit the requirements that were set at an early stage of this work. The meetings that were carried with IMAR/MARE over the time resulted in defining the following requirements:

- The system has to be easy to learn how to use;
- The system needs a user-friendly interface and experience;
- The system needs to allow free spatiotemporal data visualization;
- The system has to export a video of the changes over time;
- The system has to allow the user to visualize different variables and change the visualization parameters.

We strongly believe that we have achieved the goals of this work, by fulfilling the requirements mentioned above. We have achieved the goals by developing a browser-based application that allows the visual exploration of sediment plume dispersion in the deep-sea and the creation of animated videos suitable for dissemination and decision-making. This takes into account the requirements that were set for this work. This claim is supported by the results of the system evaluation, from both experts’ feedback and usability tests. The developed system:

- Allows the visualization of the sediment plume dispersion with color coding and flow representation techniques;
- Allows free spatio-temporal exploration;
- Includes pan, zoom and rotate camera controls;
- Allows the user to select relevant frames and collect them;
- Generates and exports video with the collected frames.
The experts’ feedback has shown that this tool is useful and allows to export good quality animations. The usability tests have shown that we have achieved the proposed goals and provided useful information about this system’s implemented features. The results of the usability tests were good, since this is a first iteration of a new tool that simplifies the process of more complex tools.

6.1 Achievements

One of the greater achievements of this work was the development of a fullstack tool that allows the spatiotemporal exploration of the dispersion of particles in the deep-sea. The frontend of this tool was developed using web-technologies. Web technologies are usually not chosen for similar use cases. We have mitigated the data loading problem by leveraging the work on the frontend with data preprocessing and loading mechanisms on the backend of the system.

Another achievement of this work is the development of a user-friendly interface and experience for the process of creating videos. This is considered an achievement due to fact that this tool mitigates the steep learning curve usually present in similar tools used for this use case.

This work achieved a solution to preprocess data, in the HDF5 file format, that is generated by the MOHID modelling system and stored. This solution consisted of using an interface to interact and manipulate the HDF5 files to export them into a more widely used format on the web. The preprocessor script was made available to help further developments. A REST API to interact with the data generated by the data preprocessor was developed. This not only facilitated the data communication between the backend and the frontend, but also empowers other applications to read and request data. This was not only useful for this work, but may be helpful for further work done in this field of research.

6.2 Future Work

Sedify can grow and improve in the future to become an even better solution for sediment plume dispersion visualization. Despite this tool being developed specifically for this use case, it is possible to imagine it being applied to many other 3D use cases. Improvements to make both the frontend and the backend more efficient, could occur in the future.

Every time the users open the application, it starts with the default configuration. A different approach for this would be to store sessions. The REST API support this by allowing the user to transfer a session and store it on the server. Later, once the user opens the application, it would be possible to load previous sessions.

The videos are currently rendered on the frontend. Different solutions for this could be explored in the future, for example, rendering the video on the backend leveraging the work that the frontend's work. Note that the current solution works as intended.

The exported videos have an overlay legend placed in one of the corners of the display. This is used to provide information of what’s being visualized. Different overlays could show along the video. For
example, annotations or descriptions that a user may want to annotate a specific part of an animation, a description of a phenomenon, some other information.

Last but not least, Virtual Reality is currently getting more used for the most varied visualization activities. As it was mentioned by one of the experts interviewed during the evaluation stage this work could take the direction of Virtual Reality. It definitely is an interesting point and worth exploring since being immersed in a video creation activity could be way more interesting.
Bibliography


[ECO14] ECORYS. Study to investigate the state of knowledge of deep-sea mining-final report to the european comission under fwc mare/2012/06—sc e1/2013/04. 2014.


Appendix A

Usability Testing Questionnaire

The questionnaire that was handed to the participants in the usability testing stage was the following one.
1. What's your age? *
   Mark only one oval.
   □ Below 25 years old
   □ Between 25 and 35 years old
   □ Between 36 and 50 years old
   □ More than 50 years old

2. What's your gender? *
   Mark only one oval.
   □ Male
   □ Female

3. I feel comfortable manipulating 3D environments. *
   Mark only one oval.
   1 2 3 4 5
   Completely disagree □ □ □ □ □ Completely agree

4. I feel comfortable using video editing tools. *
   Mark only one oval.
   1 2 3 4 5
   Completely disagree □ □ □ □ □ Completely agree

5. It was easy to identify the particles' trajectory. *
   Mark only one oval.
   1 2 3 4 5
   Completely disagree □ □ □ □ □ Completely agree

6. It was easy to pick a new date on the Timeline. *
   Mark only one oval.
   1 2 3 4 5
   Completely disagree □ □ □ □ □ Completely agree
7. It was easy to determine the video length. *
   *Mark only one oval.*
   
   1  2  3  4  5
   Completely disagree  ○  ○  ○  ○  ○  Completely agree

8. It was easy to change settings and appearance (scale, variables, colors,...) *
   *Mark only one oval.*
   
   1  2  3  4  5
   Completely disagree  ○  ○  ○  ○  ○  Completely agree

9. I think that I would like to use this system frequently. *
   *Mark only one oval.*
   
   1  2  3  4  5
   Completely disagree  ○  ○  ○  ○  ○  Completely agree

10. I found the system unnecessarily complex. *
    *Mark only one oval.*
    
    1  2  3  4  5
    Completely disagree  ○  ○  ○  ○  ○  Completely agree

11. I thought the system was easy to use. *
    *Mark only one oval.*
    
    1  2  3  4  5
    Completely disagree  ○  ○  ○  ○  ○  Completely agree

12. I think that I would need the support of a technical person to be able to use this system. *
    *Mark only one oval.*
    
    1  2  3  4  5
    Completely disagree  ○  ○  ○  ○  ○  Completely agree

13. I found the various functions in this system were well integrated. *
    *Mark only one oval.*
    
    1  2  3  4  5
    Completely disagree  ○  ○  ○  ○  ○  Completely agree
14. I thought there was too much inconsistency in this system. *
   Mark only one oval.

   1  2  3  4  5

   Completely disagree       Completely agree

15. I would imagine that most people would learn to use this system very quickly. *
   Mark only one oval.

   1  2  3  4  5

   Completely disagree       Completely agree

16. I found the system very cumbersome to use. *
   Mark only one oval.

   1  2  3  4  5

   Completely disagree       Completely agree

17. I felt very confident using the system. *
   Mark only one oval.

   1  2  3  4  5

   Completely disagree       Completely agree

18. I needed to learn a lot of things before I could get going with this system. *
   Mark only one oval.

   1  2  3  4  5

   Completely disagree       Completely agree
Appendix B

Datafiles Information

This informal document was provided by IMAR/MARE team as an explanation of the data provided.
- **Grid** -> Onde se encontra toda a geometria do modelo como latitude, longitude.
  - Bathymetry: batimetria utilizada no modelo.
  - Longitude e Latitude: geometria horizontal do modelo; coordenadas utilizadas, em neste caso, em graus.
  - ConnecktionX e ConnecktionY: longitude e latitude em metros (coordenadas esféricas).
  - VerticalZ: geometria vertical do modelo, ou seja, neste caso o modelo utilizou 50 camadas na vertical. Em cada instante do modelo estas camadas podem sofrer alteração da sua profundidade, em cada ponto i,j da malha horizontal.
  - WaterPoints3D: matriz 3d, com 1 (celula com água) e 0 (célula sem água).
- **Results** -> Onde se encontram os resultados do modelo. Os nomes das variáveis podem mudar consonte as saídas que estipulámos para o modelo. Os resultados podem ser campos 3D, 2D ou 1D ao longo do tempo.
  - No caso do primeiro ficheiro hdf5 (Lagrangian_1.hdf5), são consideradas 2 origens de partículas; (Discharge_Patorra_Clay_E e Discharge_Patorra_Clay_W) pertencentes ao mesmo grupo (Group_1) de lagrangeanos.
  - No caso do segundo ficheiro (Lagrangian_2.hdf5), são consideradas 6 origens pertencentes a 6 grupos diferentes.

Além dos resultados das partículas existem resultados comuns (Eulerianos) a todas as origens como sedimentos e Percentage Contaminated. No Group_1 encontra-se informação de saída referente às duas origens de partículas emitidas.

Em cada emissão, ou seja em cada origem, cada partícula tem a sua identificação (3D) e é seguia ao longo do tempo. Por exemplo, se se pretender fazer graficos, ao longo do tempo das partículas emitidas, teremos de escolher as variávies:

- **X pos, Y pos e Z pos** -> localização, em cada instante, em coordenadas esféricas (metros) (Z pos é a respectiva profundidade)

- **Longitude, Latitude e Z pos** -> localização, em cada instante, em graus, no nosso caso. A esta representação pode associar-se a cor que representa um determinado campo, como os sedimentos, ou temperature ou salinity ou age, etc

- **Time** -> Informação da data e hora de todo o modelo. Cada campo, do time é constituído por um array com dimensão 6, com o ano, mes, dia, hora, minuto e segundo.
Appendix C

Results Tables and Charts

Figure C.1: Distribution of the SUS questionnaire, plotted as a boxplot diagram.

Figure C.1: Distribution of the SUS questionnaire, plotted as a boxplot diagram.
Table C.1: Usability Testing. First two questions of the questionnaire.

Table C.2: Experts questionnaire with median and IQR calculated from the responses.

Table C.3: Tasks’s execution times with calculated IQR, Standard Deviation and Confidence Interval.