

Visualização 3D de manchas de óleo e partículas de microplásticos

Relatório Final do Projeto Integrador

Henrique Gardé, up202108725

José Silva, up201906576

Rúben Pereira, up202006195



Licenciatura em Engenharia Informática e Computação

Tutor: Alexandre Valle

2023/2024

Conteúdo

1. Introdução	3
1.1. Problema inicial	3
1.2. Objetivos e resultados esperados	3
1.3. Estrutura do relatório	4
2. Metodologia utilizada e principais atividades desenvolvidas	4
2.1. Metodologia apresentada	4
2.2. Intervenientes, papéis e responsabilidades	4
2.3. Atividades desenvolvidas	5
2.3.1. Planeamento de atividades ao longo do semestre	5
2.3.2. Levantamento de técnicas de renderização	6
2.3.3. Familiarização com as tecnologias a utilizar	6
3. Desenvolvimento da solução	7
3.1. Requisitos	7
3.2. Arquitetura e tecnologias	8
3.3. Solução desenvolvida	9
3.3.1. Leitura de NetCDF e conversão para STAC/JSON	9
3.3.2. Visualização de trajetórias e variáveis dependentes	12
3.3.3. Visualização de densidades	14
3.4. Validação	15
4. Conclusões	15
4.1. Resultados	15
4.2. Lições aprendidas	15
4.3. Trabalho futuro	16

1. Introdução

1.1. Problema inicial

O objetivo deste relatório é apresentar o desenvolvimento e os resultados de um projeto focado na criação de soluções inovadoras para a visualização dinâmica e realista de manchas de óleo e partículas de microplásticos. Este projeto foi realizado no âmbito do curso de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto (FEUP), em colaboração com a Faculdade de Ciências da Universidade do Porto (FCUP).

O projeto foi conduzido no contexto de investigação aplicada, com o objetivo de desenvolver ferramentas para a análise de dados ambientais, que pudessem ser utilizadas tanto em contexto académico como prático. A questão abordada diz respeito ao desafio de identificar, quantificar e visualizar eficazmente manchas de óleo e partículas de microplásticos em ambientes marinhos, que representam uma ameaça significativa para os ecossistemas e para a saúde humana. A motivação para esta investigação é a necessidade urgente de ferramentas analíticas que facilitem a análise detalhada e intuitiva destes fenómenos, contribuindo assim para a sensibilização do público e para a formulação de políticas ambientais mais eficazes.

1.2. Objetivos e resultados esperados

Este projeto tem por objetivo desenvolver soluções inovadoras para representações dinâmicas e realistas ao longo do tempo, através da criação de métodos de visualização de manchas de óleo e partículas de microplásticos, que facilitam a sua monitorização e análise.

Tendo em vista esse propósito, tivemos como objetivos operativos utilizar a técnica de voxelização para visualizar a distribuição e concentração destas substâncias em 3D, facilitando a análise espacial e temporal e desenvolver um processo robusto para transformar os dados ambientais armazenados em NetCDF num formato que facilitasse a análise dos dados. Este processo permitirá simplificar a estrutura em que os dados estão armazenados, facilitando a visualização e alargando as possibilidades de análise.

Espera-se que o projeto desenvolva um catálogo digital utilizando pySTAC, criando uma estrutura organizada e acessível para os dados convertidos em GeoJSON. Finalmente, é esperada a implementação de uma aplicação em THREE.JS que permitirá a visualização dinâmica e interativa de partículas de microplásticos e manchas de óleo no browser, oferecendo uma ferramenta intuitiva para a análise de dados.

1.3. Estrutura do relatório

O presente relatório está estruturado em várias secções, a fim de fornecer uma visão global do projeto. A *Introdução* apresenta o enquadramento do projeto, os seus objetivos e os resultados esperados. A secção *Problema inicial* descreve em pormenor as ameaças colocadas pelos microplásticos e pelos derrames de óleo, bem como os desafios na identificação e quantificação destas substâncias. A secção *Metodologia Utilizada e Principais Atividades Desenvolvidas* detalha o desenvolvimento iterativo, as técnicas de renderização e as tecnologias utilizadas no projeto. A secção *Desenvolvimento da Solução* descreve os requisitos do projeto, a arquitetura e as tecnologias utilizadas, bem como a solução desenvolvida, incluindo a leitura NetCDF, a conversão para STAC/JSON e as visualizações implementadas. Finalmente, a *Conclusão* apresenta uma avaliação dos resultados, as lições aprendidas e sugestões para trabalhos futuros.

2. Metodologia utilizada e principais atividades desenvolvidas

2.1. Metodologia apresentada

Durante o desenvolvimento deste projeto, foi adotado um método de desenvolvimento iterativo com reuniões semanais, que ocorreram através da plataforma Zoom. Nestas sessões, procedeu-se à regulação e monitorização do processo, através da verificação do trabalho realizado, esclarecimento de dúvidas e discussão do trabalho a realizar.

Para o efeito, foram utilizadas tecnologias como: o Zoom, para a interação síncrona a distância; o Gmail, como canal de comunicação em diferido entre o tutor e os autores; o Discord, como meio de comunicação entre os autores.

2.2. Intervenientes, papéis e responsabilidades

Este projeto tem como título “3D oil spill and microplastics particle visualization” (em inglês por ter sido atribuído deste modo pelo docente), com o código PE01 da Unidade Curricular Projeto Integrador 2023/2024 da L.EIC. Foi realizado pelo grupo constituído por 3 estudantes: Henrique Gardé, José Silva e Rúben Pereira. O seu acompanhamento e orientação foram assegurados pelo tutor, o professor Alexandre Valle. Perto do final da entrega do projeto, tivemos a oportunidade de apresentar o nosso trabalho à equipa do ILIAD, projeto europeu que deu contexto a este projeto integrador.

O trabalho foi estruturado do seguinte modo: Henrique Gardé desenvolveu o código de conversão de ficheiros NetCDF em ficheiros GeoJSON e STAC; José Silva e Rúben Pereira

ficaram responsáveis pela visualização de partículas. Para o desenvolvimento da visualização de volumes utilizando voxels, os três elementos trabalharam em equipa.

Quanto ao relatório, Henrique Gardé escreveu os conteúdos do ponto 2. (“Metodologia utilizada e principais atividades desenvolvidas”) e do ponto 3.3.1 (“Leitura de NetCDF e conversão para STAC/JSON”); Rúben Pereira redigiu os pontos 1. (“Introdução”), 3.1 (“Requisitos”), 3.2 (“Arquitetura e tecnologias”) e 3.4 (“Validação”); José Silva escreveu os conteúdos dos pontos 4. (“Conclusões”). José Silva e Rúben Pereira foram coautores do ponto 3.3.2 (“Visualização de trajetórias e variáveis dependentes”). Todos os elementos da equipa redigiram em conjunto o ponto 3.3.3 (“Visualização de densidades”). O relatório foi objeto de revisão final por todos os elementos do grupo de trabalho.

2.3. Atividades desenvolvidas

2.3.1. Planeamento de atividades ao longo do semestre

O semestre foi inicialmente dividido em três etapas: a primeira serviu para a exploração do problema, o levantamento de técnicas de visualização 3D e a familiarização com as tecnologias que foram utilizadas; na segunda etapa, parte do grupo desenvolveu o código de conversão de ficheiros NetCDF em ficheiros GeoJSON/STAC, enquanto a outra parte desenvolveu a visualização de partículas em 3D; a terceira etapa foi utilizada para a realização do código para a visualização de densidades, assim como a finalização do projeto. Durante a terceira etapa, apresentamos o nosso projeto ao grupo do ILIAD e foi feito e apresentado o poster do projeto. O diagrama seguinte representa o planeamento referido.

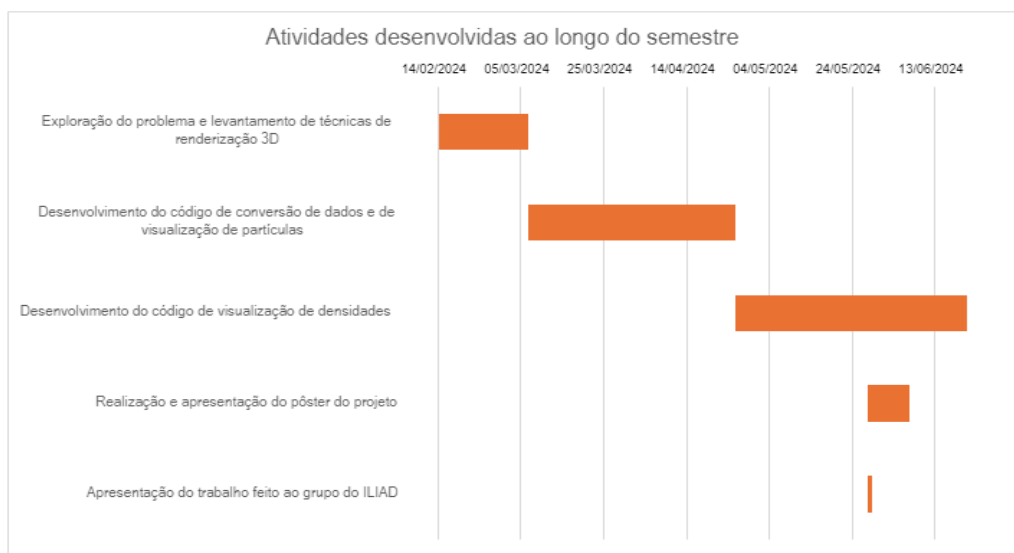


Fig. 1 - Diagrama de Gantt das atividades desenvolvidas ao longo do semestre para o Projeto Integrador

2.3.2. Levantamento de técnicas de renderização

Durante a primeira etapa, foram exploradas diversas técnicas de renderização 3D das partículas de microplásticos e das manchas de óleo. Embora no início técnicas como ray tracing e ray casting parecessem interessantes, devido à qualidade do produto final, estas foram rapidamente descartadas tendo em conta a elevada exigência quanto ao processamento das mesmas. O objetivo deste trabalho é a monitorização e análise do comportamento de partículas microscópicas. Dado o seu tamanho, a variância entre partículas que se encontram dentro de uma mesma pequena área é descartável, permitindo assim tratar e visualizar conjuntos de partículas de microplásticos, ao invés de partículas de microplásticos individuais. Após analisar todos estes fatores, escolhemos o uso de voxels, pixels com um novo parâmetro espacial que lhes permite representar o volume, e consequentemente estudar as densidades dos objetos que os ocupam. A manipulação dos voxels de acordo com diferentes opções de visualização favorece uma melhor análise e estudo das partículas de microplásticos presentes nos oceanos, afetando minimamente a performance quando comparado com outros métodos de renderização 3D.

2.3.3. Familiarização com as tecnologias a utilizar

A familiarização com as tecnologias foi um passo essencial para garantir o correto desenvolvimento da solução, tendo havido um período de aprendizagem e experimentação das ferramentas escolhidas.

A conversão de dados para diferentes ficheiros exigiu um estudo dos diferentes tipos de formatos envolvidos neste projeto. O formato de dados em que os dados iniciais se encontram chama-se NetCDF. Este é muito utilizado para armazenar dados multidimensionais e tem uma estrutura complexa que requer uma excelente compreensão por parte de quem a utiliza. Outro formato que foi necessário estudar foi GeoJSON, formato baseado em JSON, utilizado para a representação de recursos geográficos simples, juntamente com os seus atributos não espaciais. Familiarizamos-nos com bibliotecas Python para a manipulação de ficheiros NetCDF, como *netCDF4-python* e *pySTAC*. *pySTAC* é uma biblioteca usada para criar e gerenciar catálogos STAC (SpatioTemporal Asset Catalog), que nos permite referenciar ficheiros, neste caso, os ficheiros GeoJSON com a informação vinda dos ficheiros NetCDF, e armazenar outros tipos de informação tal como os limites espaciais e o intervalo de tempo dos ficheiros referenciados. Realizamos exercícios práticos de leitura, processamento e conversão de dados NetCDF para formatos mais acessíveis como GeoJSON.

Para o desenvolvimento da parte de visualização do projeto foi utilizado WebGL, uma API em Javascript para a renderização de gráficos 2D e 3D. Como todos os membros do grupo tiveram Computação Gráfica no 3.º ano da Licenciatura em Engenharia Informática e Computação, no âmbito da qual foi trabalhada esta biblioteca, todos tínhamos conhecimento sobre ela. Foi ainda investido tempo para nos familiarizarmos com a biblioteca THREE.JS, uma framework de Javascript que utiliza WebGL, que auxilia a demonstração de gráficos 3D

animados. Realizamos tutoriais de pequenos projetos para entender melhor a manipulação de câmaras, luzes, materiais e objetos. Foram explorados exemplos de visualização de partículas e volumes disponíveis na documentação do THREE.JS.

3. Desenvolvimento da solução

3.1. Requisitos

Para um desenvolvimento tranquilo e isento de dificuldades foi necessário estar consciente dos requisitos funcionais e não funcionais. Nesta secção, explicitamos alguns destes requisitos, tais como restrições ao projeto. Dividimos esta explicitação em requisitos funcionais, não funcionais e restrições.

Quanto aos requisitos funcionais, importa referir, pelo menos, três.

- **Conversão de dados.** O usuário deve conseguir converter os dados originais que estão em formato NetCDF para um formato de ficheiro mais fácil de interpretar e analisar. É necessário que o código seja corrido usando a linha de comandos de forma a possibilitar ao usuário escolher a opção de conversão mais adequada ao ficheiro inicial e às necessidades do mesmo.
- **Visualização de partículas.** O usuário deve ser capaz de visualizar partículas de microplásticos e manchas de óleo em 3D. Devem ser representados tanto a distribuição das partículas como os seus movimentos ao longo do tempo. É necessário mostrar informações adicionais sobre as partículas, tais como a trajetória e os vetores que representam a velocidade.
- **Visualização de densidades.** O usuário deve conseguir visualizar os voxels que contêm partículas. Deve ser clara a localização dos voxels tal como o que está no seu interior de forma a que o usuário consiga interpretar corretamente o que está a ver.

Quanto aos requisitos não funcionais, mencionamos os três seguintes.

- **Prevenção de erros na conversão.** O código de conversão deve garantir que existem todos os dados fornecidos pelo utilizador. Caso não existam (imagine-se, por exemplo, o nome dos ficheiros que têm que ser convertidos têm um erro), o código deve parar de correr e deve avisar o utilizador de que há um erro nos dados introduzidos.
- **Desempenho.** O sistema deve ser capaz de processar grandes volumes de dados de forma eficiente, garantindo uma experiência fluida.
- **Interatividade.** O sistema deve permitir a interação do usuário, de forma a promover a mudança da perspectiva de visualização. Deve ser possível desligar

e ligar qualquer aspeto da visualização tal como alterar os tamanhos, por exemplo, das partículas. Este requisito tem em mente a experiência do utilizador.

Quanto às restrições, há duas a mencionar.

- **Tempo.** Um dos motivos que nos causou mais transtorno foi o tempo. Conciliar todas as unidades curriculares com o Projeto Integrador foi desafiante, dado que grande parte das aprendizagens das UC também envolviam desenvolver projetos. Este desafio resultou, de facto, numa oportunidade para desenvolvermos as nossas competências de gestão de processos.
- **Informação sobre as tecnologias utilizadas.** Apesar de haver bastante informação na internet (por exemplo, tutoriais) acerca de algumas das tecnologias específicas do projeto, a informação sobre certos formatos de ficheiros era pouca e não muito clara. Consideramos que este fator nos limitou um pouco no que conseguimos fazer, pelo menos no início do projeto.

3.2. Arquitetura e tecnologias

O projeto foi realizado através da utilização de uma série de ferramentas e bibliotecas, que passamos a enunciar.

- **Python.** Utilizado para a conversão de dados, o Python foi escolhido devido à sua vasta biblioteca de ferramentas de manipulação de dados e ao suporte para NetCDF e GeoJSON.
- **THREE.JS.** Esta biblioteca JavaScript foi utilizada para a visualização 3D no navegador, permitindo a criação de gráficos interativos e dinâmicos. A biblioteca THREE.JS é amplamente reconhecida pela sua eficiência e flexibilidade na apresentação de gráficos 3D na Web.
- **GeoJSON.** Um formato baseado em JSON para codificar uma variedade de estruturas de dados geoespaciais. O GeoJSON foi escolhido pela sua interoperabilidade e facilidade de integração com ferramentas de cartografia e visualização.
- **NetCDF.** É um formato de dados utilizado para armazenar dados multidimensionais, tais como variáveis climáticas e oceanográficas. Os dados a visualizar estão no formato NetCDF, que foi estudado e compreendido para efeitos do projeto.
- **pySTAC.** É uma biblioteca Python utilizada para criar e manipular catálogos de ativos espaço-temporais (STAC). A sua implementação facilitou a organização e acessibilidade dos dados convertidos em GeoJSON.

O diagrama (fig.2) a seguir apresentado representa as componentes e o modo de interação entre elas. É possível observar que, no início, é feita a conversão dos ficheiros NetCDF para ficheiros GeoJSON. Esses novos ficheiros são referenciados pelo catálogo criado com a STAC. Para a visualização de partículas e de densidades, os dados utilizados são os que se encontram nos ficheiros GeoJSON. Todas as diferentes etapas necessitam de um utilizador, isto é, tanto na conversão como na visualização das partículas e densidades, é necessário que o utilizador corra manualmente os diferentes códigos.

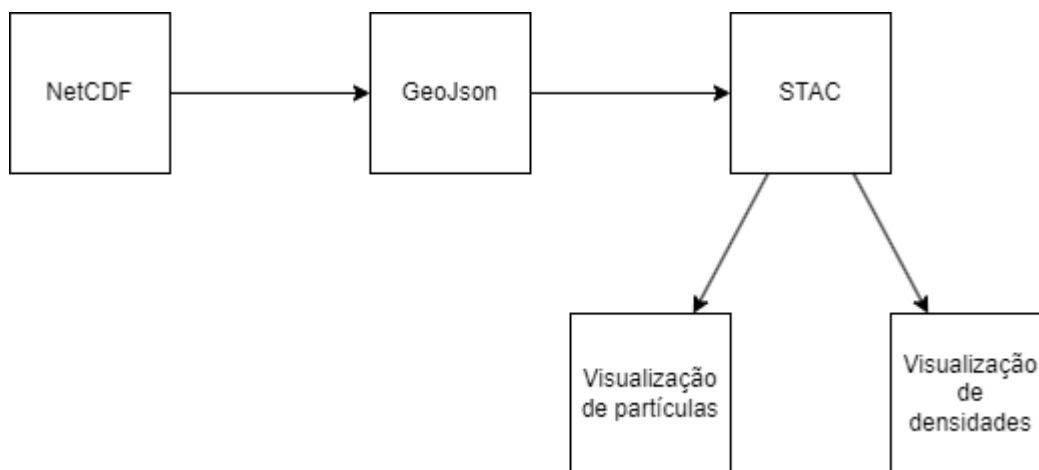


Fig. 2 - Diagrama de Componentes

Na prossecução do projeto, deparámos com duas dificuldades técnicas. Por um lado, o desconhecimento inicial da biblioteca THREE.JS, o que representou um desafio no desenvolvimento das visualizações 3D. Por outro lado, a visualização de grandes volumes de dados geraram problemas de desempenho e de lentidão.

3.3. Solução desenvolvida

3.3.1. Leitura de NetCDF e conversão para STAC/JSON

Embora a conversão de ficheiros do formato NetCDF para GeoJSON/STAC não seja o objetivo principal deste projeto, é uma etapa importante para a agilização do processo. Assim, de forma a facilitar a leitura dos dados fornecidos, desenvolveu-se este conversor de ficheiros.

O formato NetCDF é muito utilizado no armazenamento de dados utilizados em aplicações de climatologia, meteorologia e oceanografia devido ao suporte para grandes quantidades de dados e dados multidimensionais, isto é, dados que variam com outra variável, por exemplo, o tempo ou o espaço. Cada ficheiro contém metadata que descreve a informação armazenada, tal como nomes de variáveis, unidades de cada variável e dimensões. Para

informação em 3D, a informação é armazenada como uma série de matrizes bidimensionais. Estes fatores tornam o ficheiro mais fácil de ler por um utilizador, mas a complexidade do armazenamento revela-se um desafio para a utilização da informação, no nosso caso, na visualização dessa informação. A estrutura de um ficheiro NetCDF está representada na figura 3. Dada a complexidade do formato, consideramos conveniente fazer um conversor de ficheiros NetCDF para ficheiros GeoJSON, facilitando a leitura da informação por parte do programa que trata da visualização de partículas e densidades.

```
netcdf filename {
dimensions:
    lat = 1;
    lon = 2;
    time = UNLIMITED; // (2 currently)

variables:
    float lat(lat) ;
        lat:long_name = "Latitude" ;
        lat:units = "degrees_north" ;
    float lon(lon) ;
        lon:long_name = "Longitude" ;
        lon:units = "degrees_east" ;
    int time(time) ;
        time:long_name = "Time" ;
        time:units = "days since 2024-02-13" ;
    float temperature(time, lat, lon) ;
        temperature:long_name = "Temperature" ;
        temperature:units = "degrees_celsius" ;
data:
    lat = 40.0;
    lon = -100.0, -120.50;
    time = 500, 800;
    temperature = 20, 20.5, 19.65, 16.3;
```

Fig. 3 - Estrutura do NetCDF

O formato GeoJSON é baseado em JSON e foi projetado para representar recursos geográficos simples, juntamente com os seus atributos não espaciais. Converter os dados para o formato GeoJSON permite-nos formatar os dados da forma que o utilizador considera mais adequada, facilitando o tratamento de dados e o seu uso para a visualização dos mesmos.

A utilização de um catálogo criado com a biblioteca pySTAC revela-se útil na leitura dos diferentes ficheiros GeoJSON. Cada ficheiro NetCDF contém diversos instantes de tempo e várias coordenadas espaciais. É possível assim gerar diversos ficheiros GeoJSON diferentes para apenas um ficheiro inicial. De forma a organizar os diversos ficheiros de output, foi utilizado um catálogo que contém referências, isto é, o caminho relativo dos ficheiros, para todos os ficheiros gerados.

O código é corrido pelo terminal e o utilizador tem que passar os diferentes argumentos necessários, sendo estes: o nome do ficheiro em formato NetCDF que pretende converter; o nome que será utilizado como base do nome dos ficheiros de output, o nome do catálogo que vai ser criado, os diferentes tipos de conversão que podem ser feitos, os diferentes tipos de output, o nome da variável de tempo e os nomes das variáveis de espaço. Também podem ser passados como argumento outros tipos de dados como os nomes dos diferentes vetores de velocidade e nomes de outras variáveis relevantes. A conversão pode ser feita de dois modos diferentes: *explicit*, os valores armazenados correspondem aos valores de todas as partículas nos diferentes instantes de tempo; *calculated*, os valores armazenados correspondem aos valores de todas as partículas apenas no instante inicial. Os valores das coordenadas espaciais das partículas nos restantes instantes é calculado usando o tempo e as velocidades. Para além disso, o utilizador pode escolher entre diferentes modos de output: *time*, cada ficheiro representa um instante e contém os diferentes valores de cada partícula naquele instante; *particle*, cada ficheiro representa uma partícula e contém todos os dados relativos a cada partícula nos diferentes intervalos de tempo.

Na imagem abaixo é possível ver um excerto de um ficheiro GeoJSON que representa uma partícula.

```
1 {
2   "type": "FeatureCollection",
3   "Particle number": 1,
4   "features": [
5     {
6       "type": "Feature",
7       "geometry": {
8         "type": "Point",
9         "time": 0.0,
10        "coordinates": [
11          -50.62530547391073,
12          -85.9275222066742,
13          -7999.139572458407
14        ]
15      },
16      "property": {
17        "velocity 1": [
18          0,
19          0,
20          0
21        ],
22        "velocity 2": [
23          0,
24          0,
25          0
26        ],
27        "lat": -50.62530547391073
28      }
29    },
30  ]
31 }
```

Fig. 4 - Excerto de um ficheiro GeoJSON que representa uma partícula

3.3.2. Visualização de trajetórias e variáveis dependentes

A biblioteca THREE.JS é utilizada no processo de visualização, para criar gráficos interativos e dinâmicos no browser. A correta representação de trajetórias e variáveis dependentes é um aspeto fundamental para compreender o comportamento espacial e temporal das partículas de microplásticos. Esse processo é composto por várias etapas que iremos explicar nos parágrafos seguintes.

É importante referir que, como existem milhões de partículas de microplásticos nos oceanos, seria impossível renderizá-las todas. Assim, quando nos referimos a partículas neste relatório, estamos a designar um agrupamento de partículas. Dessa forma, é possível demonstrar todas as partículas de uma forma aproximada com uma maior performance.

A fase inicial consiste na importação e no tratamento dos dados relativos à trajetória, que representam o movimento das partículas. Estes incluem as três coordenadas espaciais: latitude, longitude e profundidade; e a variável tempo. Os dados são carregados a partir dos ficheiros GeoJSON utilizando uma função chamada *loadCoordinates*, que vai buscar os dados espaciais referentes a cada partícula e os coloca numa matriz de coordenadas representativas das posições de cada partícula ao longo do tempo. Estes dados são renderizados num ambiente tridimensional utilizando THREE.JS, o que permite a sua visualização. Um dos problemas que surgiu durante o desenvolvimento da renderização de partículas foi a fluidez da animação. Se apenas utilizássemos os dados que se encontram nos ficheiros, as partículas, em vez de se moverem no espaço ao longo do tempo, iriam mudar de posição apenas nos instantes que se encontravam nos ficheiros. De forma a resolver esse problema, implementamos a interpolação linear dos valores das coordenadas espaciais.

São inicializados uma cena, uma câmara e um renderizador para estabelecer o espaço 3D no qual as partículas serão visualizadas, como representado na figura 6. As partículas são criadas utilizando os objetos *BufferGeometry* e *PointsMaterial*, e são adicionadas à cena. Os utilizadores podem navegar no espaço tridimensional e observar as partículas de diferentes ângulos e perspetivas.

A interatividade do utilizador com a visualização das partículas e densidades é uma componente crucial, permitindo aos utilizadores interagir com as visualizações através de uma interface gráfica do utilizador (GUI). A GUI, como representado na figura 5, inclui uma série de ferramentas que permitem ao utilizador aumentar o tamanho das partículas, mudar a transparência das mesmas e ligar ou desligar diferentes componentes da visualização tais como a visibilidade de eixos, partículas e trajetórias. Estas diferentes opções permitem ao utilizador realizar explorações detalhadas do ambiente 3D.

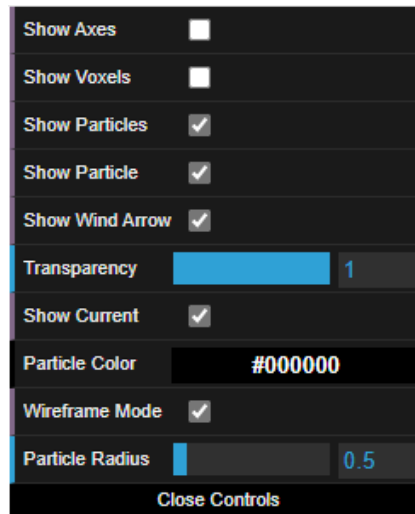


Fig. 5 - Interface gráfica

As trajetórias representam o movimento da partícula desde o instante inicial até ao instante atual. Elas são coloridas para proporcionar uma visualização mais intuitiva das alterações ao longo do tempo e do espaço. A função que trata da animação é responsável por atualizar as posições das partículas e renderizar a cena, permitindo assim aos utilizadores observar o comportamento dinâmico das partículas.

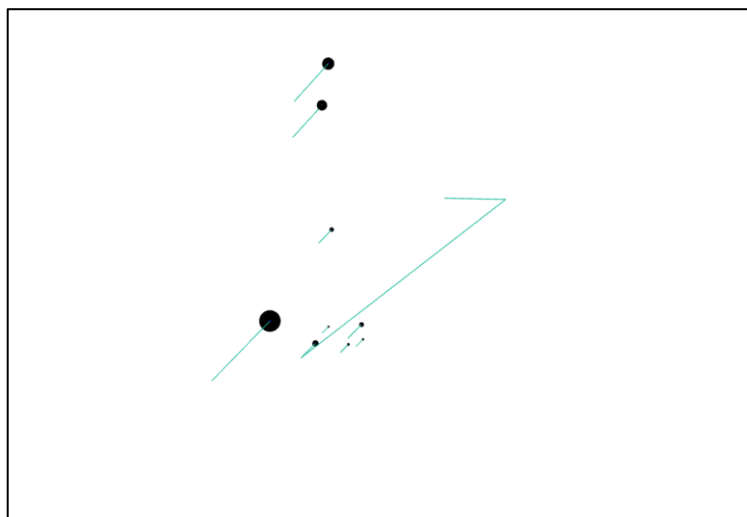


Fig. 6 - Visualização das partículas e das suas trajetórias

3.3.3. Visualização de densidades

A visualização das densidades é uma componente crucial da aplicação que permite uma análise mais detalhada da distribuição espacial das partículas. Utilizamos a técnica de voxelização para representar essas densidades em um espaço tridimensional. Um voxel é um cubo tridimensional que contém informações sobre a concentração de partículas nessa área específica.

Na nossa aplicação, os voxels são utilizados para dividir o espaço tridimensional em pequenas células que armazenam a contagem de partículas dentro do seu volume. Para criar a grade de voxels, primeiramente determinamos os limites máximos e mínimos do espaço tridimensional a partir dos dados carregados. Com base nesses limites, calculamos a dimensão do espaço e dividimos essa área em uma grade tridimensional composta por voxels de tamanho uniforme.

Os voxels são responsáveis por contar o número de partículas nas suas respectivas áreas durante a simulação. Conforme as partículas se movem pelo espaço, a contagem em cada voxel é atualizada. Quando uma partícula se move de um voxel para outro, a contagem de partículas do voxel anterior decrementa e é incrementada a contagem no novo voxel. Esta contagem dinâmica permite que a aplicação monitore a densidade de partículas em cada voxel.

O processo de contagem é gerenciado pela classe *VoxelManager*. Conforme o movimento das partículas, a posição delas é usada de forma a ser calculado o índice do voxel em que essa partícula se encontra. Este índice é calculado em cada frame da animação e é necessário saber se o índice atual é diferente ou igual ao índice associado à partícula no frame anterior. Para isso, guardamos sempre o índice do frame anterior para cada partícula. Caso seja diferente, o valor da contagem do voxel anterior diminui, enquanto a contagem do novo voxel aumenta. Caso os índices sejam iguais, não existem subtrações e adições. Este método permite-nos ter dois modos de visualização diferentes: contagem atual, que apenas mostra os voxels que contêm partículas; contagem acumulada, que mostra todos os voxels que já tiveram e que contêm partículas.

Em resumo, a voxelização permite uma representação clara e detalhada da densidade de partículas em cada voxel, oferecendo uma visão simplificada da distribuição espacial e temporal das partículas.

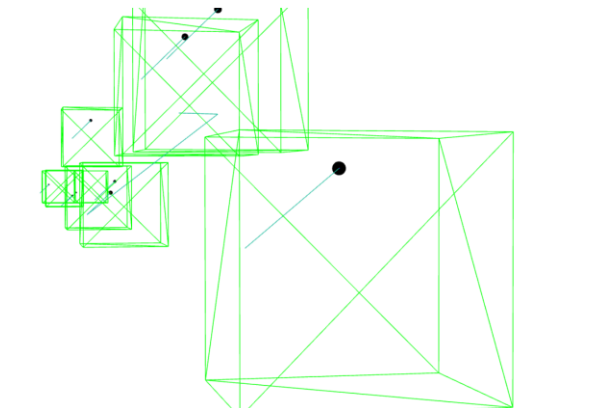


Fig. 7 - Visualização de densidades com a opção contagem atual

3.4. Validação

Ao longo do semestre, o projeto foi sendo validado pelo tutor e, numa reunião perto do final do semestre, pelo grupo do ILIAD, garantindo assim que os objetivos estavam a ser cumpridos. Este processo de validação contínua é fundamental para a evolução do projeto com a adição de novas funcionalidades.

4. Conclusões

As conclusões que derivam do exposto neste relatório são organizadas em resultados, lições aprendidas e trabalho futuro.

4.1. Resultados

Com o desenvolvimento deste projeto, obteve-se uma aplicação, construída em THREE.JS, capaz de visualizar partículas num ambiente tridimensional. Esta ferramenta oferece uma visão detalhada e dinâmica da distribuição e movimento dessas partículas ao longo do tempo. Além disso, implementamos uma funcionalidade de conversão de dados NetCDF para o formato GeoJSON e a criação de um catálogo digital utilizando pySTAC, simplificando o processo de tratamento e análise dos dados.

Quanto ao processo cooperativo e colaborativo de realização do projeto, avaliamos os contributos individuais como tendo sido um bom trabalho, tendo cada um intervindo com 33% do esforço individual tanto no projeto como no relatório.

4.2. Lições aprendidas

Durante o desenvolvimento do projeto, desenvolvemos competências de gestão de projeto, resolução de problema, de pensamento crítico e criativo, de negociação de contributos individuais, além de termos adquirido conhecimentos sobre diversas novas tecnologias. A utilização de THREE.JS para a criação de visualizações 3D mostrou ser uma escolha acertada, proporcionando o desempenho adequado para as nossas necessidades. A implementação da biblioteca pySTAC foi essencial para a organização e gerenciamento dos diferentes ficheiros GeoJSON. Outra aprendizagem importante foi o estudo de novas tecnologias. O uso de tecnologias inicialmente desconhecidas para nós ensinou-nos como estudar e implementar as mesmas de forma a cumprir os objetivos. Experimentamos mais uma vez a necessidade de ter um bom plano antes de começar o desenvolvimento de um projeto. Finalmente, tivemos a oportunidade de melhorar as nossas habilidades de comunicação e colaboração entre membros de equipa e o tutor.

4.3. Trabalho futuro

No futuro, este projeto pode seguir várias direções para aprimorar ainda mais a aplicação desenvolvida. Uma área de melhoria é a interface do usuário, que pode ser ainda mais amigável e funcional, incluindo ferramentas adicionais e diferentes modos de visualização. Outro aspecto que pode ser melhorado é a arquitetura. Devido ao código relativo à conversão ser feito pelo terminal, é possível automatizar esse processo. É possível então transformar a arquitetura deste projeto em servidor-cliente, em que o servidor trata da conversão de dados e da visualização dos dados tratados, e o cliente apenas necessita de se ligar ao servidor e utilizar a GUI para visualizar e analisar os dados da forma que achar mais adequada. Atualmente, a nossa aplicação enfrenta limitações ao processar grandes conjuntos de dados, o que pode comprometer a fluidez e a interatividade da visualização 3D. Para superar este desafio, podemos optar por técnicas de processamento paralelo e otimização de algoritmos. Outra implementação valiosa seria a utilização de cores usadas como escala para cada voxel mediante a quantidade de partículas contidas no voxel. Essa abordagem permitiria que os usuários identificassem facilmente áreas de maior concentração de partículas, tornando a análise visual mais intuitiva e detalhada. A aplicação de gradientes específicos pode realçar visualmente regiões com maior interesse, facilitando a identificação de padrões nos dados. Ao incorporar estas melhorias à aplicação, não apenas aumentará a sua capacidade de gerenciamento de dados, mas também se tornará uma ferramenta mais eficaz para a análise de dados.