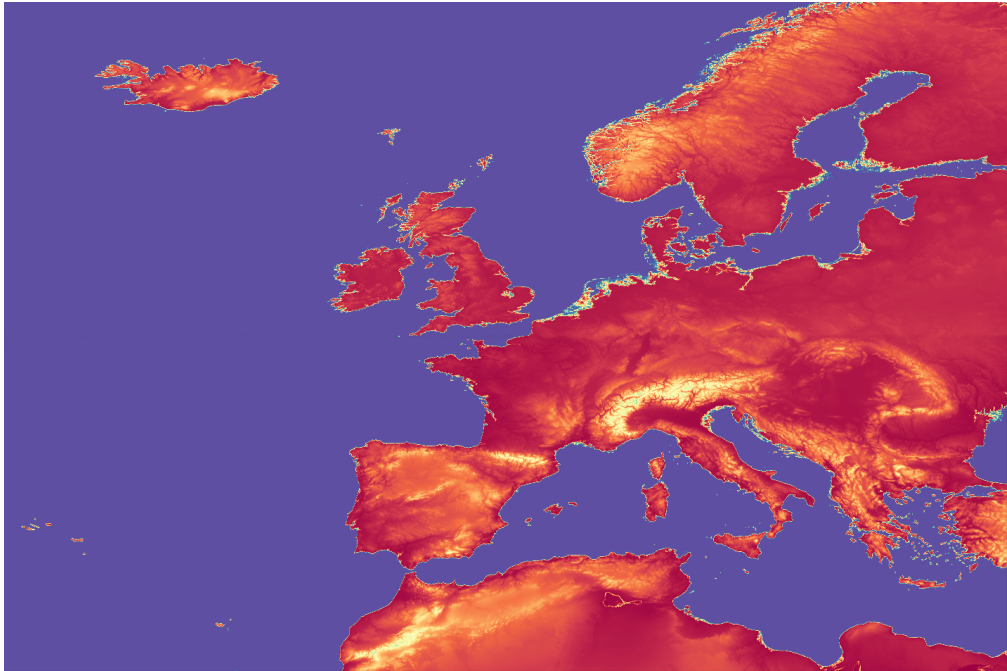


TrackFinder en détail



Représentation des données de la carte a l'heure actuelle.

Tech-Stack

Back-end:

- Python
- Flask

Base de données:

- Postgres
- Adminer

Front-End:

- A déterminer entre React/Angular/Vue pour le site
- A déterminer pour la version mobile

Déploiement:

- Docker
- Docker Compose

A noter qu'un prototype du back-end et de déploiement est déjà disponible.

Les bibliothèques principales sont Rasterio pour la lecture des fichiers .tif, psycopg2 pour la gestion des requêtes SQL et Numpy pour la gestion d'arrays rapide.

Une copie de la base de données pourra être partagée via un système de dépôt de fichiers, la création initiale de celle-ci pouvant être assez longue selon la puissance de l'ordinateur.

Le front-end reste à définir, en fonction des compétences des membres du groupe et des capacités des langages par rapport à ce qui va être demandé.

Adminer est mis en place pour surveiller la base de données, mais n'est pas obligatoire à 100% et peut évidemment être changé en cas de besoin, c'est plus un outil de confort pour les vérifications qu'une nécessité.

Sources des données

- GMTED2010: Contient les données topographiques de toute la planète organisées en rectangles de 30x20.
https://topotools.cr.usgs.gov/gmted_viewer/viewer.htm
- Copernicus: Données géographiques des forêts.
<https://land.copernicus.eu/pan-european/corine-land-cover/clc2018?tab=download>
- EPSG.io: Permet de convertir les coordonnées des cartes en longitude et latitude en position sur la carte.
https://epsg.io/transform?s_srs=4326&t_srs=3857&x=NaN&y=NaN

A noter que pour GMTED2010, ne sont utilisés que les 4 rectangles de l'Europe, qui seront sous la forme de fichiers .tif, où chaque pixel de l'image représente 250m².

Algorithme utilisé

L'algorithme utilisé est basé sur le A*, l'un des plus connus et utilisés de nos jours en raison de sa rapidité et de la quantité de mémoire consommée relativement faible, notamment comparé au Dijkstra.

Celui-ci fonctionne avec un système de queue ou de file d'attente, dans laquelle chaque emplacement y est classé selon le résultat d'une fonction (déterminée par le programmeur selon ses besoins et attentes).

De manière simplifiée, celui-ci marche de la façon suivante:

- Etape 1: On détermine le point de départ et de fin.
- Etape 2: Le point de départ devient le point actuel.
- Etape 3: On regarde et enregistre les voisins de la position actuelle.
- Etape 4: Les points enregistrés sont classés en fonction d'un score déterminé en regardant la distance parcourue depuis le point de départ et la distance restante jusqu'à l'arrivée.
- Etape 5: Le point actuel devient celui avec le score le plus bas.

- On répète les opérations depuis l'étape 3 à 5 jusqu'à atteindre la fin.

Cet algorithme est privilégié car permettant de ne visiter, sauf exceptions ou terrains atypiques, que les points de la carte se rapprochant de la fin, et privilégie les endroits où le trajet est le moins coûteux.

Une explication pratique peut être trouvée dans cette vidéo, en anglais mais très bien expliquée et démonstrative:



Déploiement en 3 parties

- Lancement des containers via docker compose (Postgres, Adminer, Python).
- Pré-processing de la data puis lancement de l'API.
- Communication avec le front-end.

<input type="checkbox"/>	t-esp-700-starters-28612-track_finder-0	-	Running (2/3)	
<input type="checkbox"/>	python_worker 3891884fd594	t-esp-700-starters-28612-track_finder-0-python_worker:latest	Exited (1)	
<input type="checkbox"/>	adminer-1 81f42291053a	adminer:latest	Running 8080:8080	10 minutes ago
<input type="checkbox"/>	postgres_db 593728b72906	postgres:latest	Running 8000:5432	10 minutes ago

Détails du pré-processing.

Explication des fichiers Geotiff:

Les fichiers Geotiff sont des images, mais aussi et surtout des immenses arrays de plusieurs couches de données.

La librairie Rasterio récupère ces informations et les organise en array Numpy, beaucoup plus rapide a manipuler que celles de Python de base. Elle récupère également les meta-datas de celle-ci, où se trouve notamment les coordonnées géographiques des zones ainsi que le format de coordonnées utilisées.

Les fichiers obtenus via GMTED2010 sont des rectangles de 9600 x 14400 pixels, avec l'altitude moyenne pour chaque pixel (250m²). Seront cependant ignorés ceux dont l'altitude est égale à 0, car représentant l'océan, et donc inutilisables pour les transport à pied.

L'enregistrement des données se fait en 3 étapes:

- La création des nodes:
Pour chaque pixel des images, on attribue une position X et Y comme sur un repère en 2 dimensions (c'est une sorte d'immense carte après tout) afin d'avoir sa position. Sont enregistrées tout les nodes d'altitude supérieures a 0.
- Liaison des nodes:
On observe chaque node dans la base de données et on lui enregistre ses voisins dans leur position respective, afin de savoir dans le futur qui est lié à quoi.
Il s'agit de la partie la plus longue du processus.
- Troisième étape:
Il est impossible de se déplacer dans l'océan ou les mers, donc un départ en Angleterre pour arriver en Europe à pied représente une requête à laquelle il n'y a pas de solution, et qui prendra énormément de temps et de ressources à résoudre, pour au final n'avoir aucun résultat.
Un système de zone est donc créé, pour savoir quels groupes de nodes sont isolés les uns des autres, et où la communication est impossible avec l'extérieur (Entre autres les îles). Pour les trouver, on forme une sorte de pseudo vague, où on prend une node, on regarde les voisins, et les voisins des voisins etc jusqu'à ne plus avoir de voisins directs existants et on leur attribue à tous une zone. On peut faire la même chose avec la node suivante dans la base de données qui n'a pas encore de zone attribuée.

Ce système de pré-processing permet de gagner un temps considérable lors du lancement d'algorithmes, ou bien pour filtrer les requêtes impossibles d'offices.
De plus il permet une économie de ressources considérable, car avoir toutes les nodes en mémoire nécessiterait une quantité de RAM considérable sans pour autant obtenir de meilleurs résultats.

<input type="checkbox"/> Modify	id	x_pos	y_pos	north_neighbor	north_east_neighbor	east_neighbor	south_east_neighbor	south_neighbor	south_west_neighbor	west_neighbor	north_west_neighbor	elevation_value	zone_id
<input type="checkbox"/> edit	1	5469	1382	NULL	NULL	2	5	4	NULL	NULL	NULL	1	1
<input type="checkbox"/> edit	2	5470	1382	NULL	NULL	3	6	5	4	1	NULL	2	1
<input type="checkbox"/> edit	3	5471	1382	NULL	NULL	NULL	7	6	5	2	NULL	1	1

<input type="checkbox"/> edit	4	5469	1383	1	2	5	NULL	NULL	NULL	NULL	NULL	3	1
<input type="checkbox"/> edit	5	5470	1383	2	3	6	NULL	NULL	NULL	4	1	6	1
<input type="checkbox"/> edit	6	5471	1383	3	NULL	7	NULL	NULL	NULL	5	2	2	1
<input type="checkbox"/> edit	7	5472	1383	NULL	NULL	NULL	NULL	NULL	NULL	6	3	1	1
<input type="checkbox"/> edit	8	5751	1649	NULL	NULL	NULL	11	10	9	NULL	NULL	2	2
<input type="checkbox"/> edit	9	5750	1650	NULL	8	10	NULL	NULL	NULL	NULL	NULL	1	2
<input type="checkbox"/> edit	10	5751	1650	8	NULL	11	13	NULL	NULL	9	NULL	10	2
<input type="checkbox"/> edit	11	5752	1650	NULL	NULL	12	14	13	NULL	10	8	12	2
<input type="checkbox"/> edit	12	5753	1650	NULL	NULL	NULL	15	14	13	11	NULL	1	2
<input type="checkbox"/> edit	13	5752	1651	11	12	14	16	NULL	NULL	NULL	10	8	2
<input type="checkbox"/> edit	14	5753	1651	12	NULL	15	17	16	NULL	13	11	14	2
<input type="checkbox"/> edit	15	5754	1651	NULL	NULL	NULL	18	17	16	14	12	3	2
<input type="checkbox"/> edit	16	5753	1652	14	15	17	21	20	NULL	NULL	13	27	2
<input type="checkbox"/> edit	17	5754	1652	15	NULL	18	22	21	20	16	14	52	2
<input type="checkbox"/> edit	18	5755	1652	NULL	NULL	19	23	22	21	17	15	28	2
<input type="checkbox"/> edit	19	5756	1652	NULL	NULL	NULL	24	23	22	18	NULL	3	2
<input type="checkbox"/> edit	20	5753	1653	16	17	21	29	28	27	NULL	NULL	13	2
<input type="checkbox"/> edit	21	5754	1653	17	18	22	30	29	28	20	16	61	2
<input type="checkbox"/> edit	22	5755	1653	18	19	23	31	30	29	21	17	79	2
<input type="checkbox"/> edit	23	5756	1653	19	NULL	24	32	31	30	22	18	64	2
<input type="checkbox"/> edit	24	5757	1653	NULL	NULL	NULL	33	32	31	23	19	20	2
<input type="checkbox"/> edit	25	5750	1652	NULL	NULL	26	26	25	24	NULL	NULL	1	2