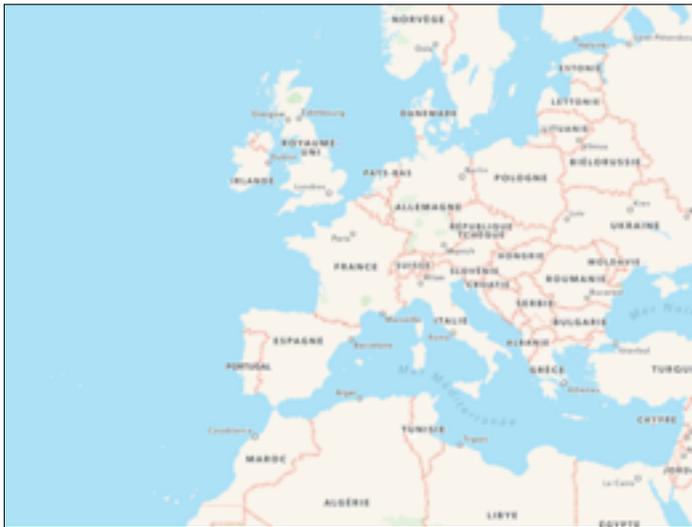


# POI Clustering considered harmful



Nicolas Bouil-Boud  
IOG @ Captain Train  
@\_nb

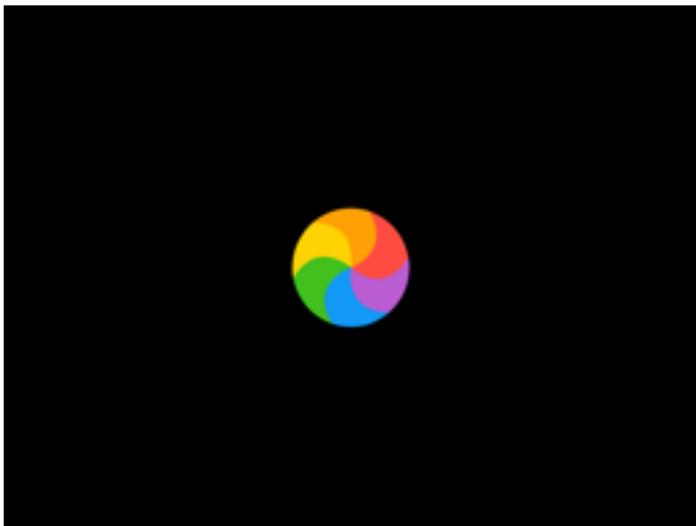
POI: point of interest. Les points affichés sur une carte.



Captain Train a récemment publié en opendata un fichier “stations.csv” contenant une liste de gares européennes de différents transporteurs.

20000 stations.

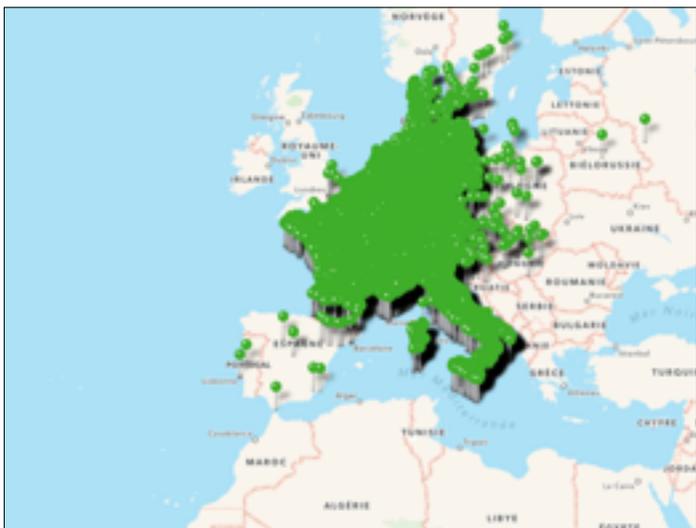
Quand on charge 20000 POI dans mapkit, on à d'abord ça.

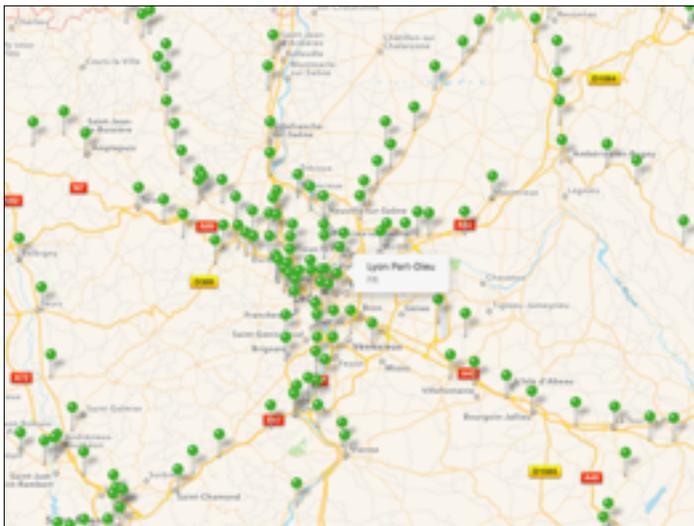


Puis ça.

3 problèmes:

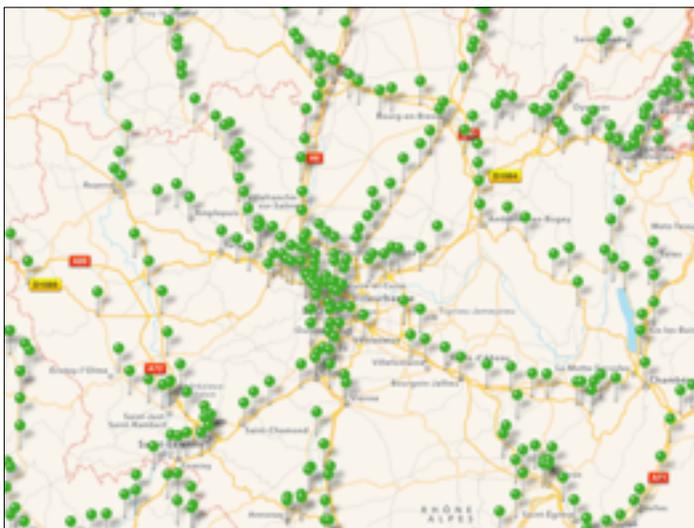
- \* Performance: lags, mémoire, contrôle presque impossible.
- \* Usabilité: on ne peut pas sélectionner chaque item indépendamment
- \* Information: tellement de POIs qu'on ne voit plus rien entre eux.





De près, ça marche assez: on voit chaque point indépendamment, on arrive à les sélectionner, et ça scrolle bien, sans saccade.

(Remarque: même en ajoutant le 20000 POI à la MKMapView. MapKit est suffisamment malin pour ne pas prendre en compte les POIs hors-champ, mais UIKit tombe quand il doit afficher 20000 vues les unes par dessus les autres.)



En dézoomant juste un peu, ça redevient difficile: performance, usabilité.

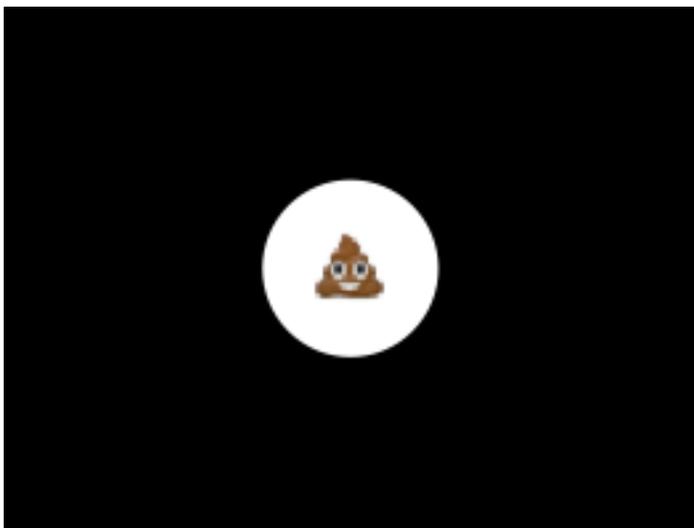
La solution courante, consiste à regrouper les POIs proches ensemble. Il existe plusieurs algos de "POI Clustering". il y a eu pas mal de blogposts sur le sujet, surtout il y a quelques années, quand même afficher 100 POI sur un iPhone était difficile.

(Remarque: Aujourd'hui, sur un iPhone récent, afficher 100 POI ne pose aucun problème de performance, du coup les développeurs se contentent de ça, sans régler les autres problèmes.)



Plusieurs algos: k-means, précis, mais lent, ou des techniques de quadrees, ce que font la plupart des libs open-source.

Ça donne ça: chaque petit railtag sur la carte est placé au barycentre de plusieurs stations proches. Notons qu'on voit encore quelques stations toutes seules. Ça règle les problèmes de performance et d'usabilité. Youpi!

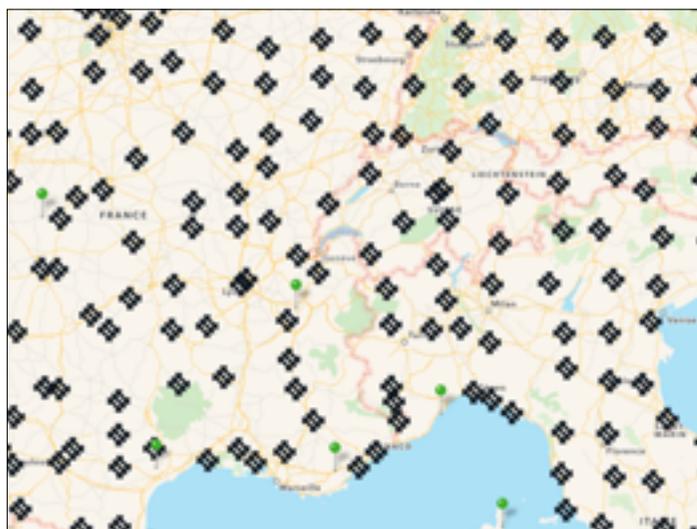


Et en fait, c'est une mauvaise solution.

On affiche n'importe quoi à la place des POI, et on rajoute d'autres bugs.



Premier effet pervers des quadtree: un effet “grille”, ou les POIs cluster sont tous alignés.



On voit particulièrement bien la grille ici.

Deuxième problème: instabilité. les POIs ne se regroupent pas toujours exactement de la même façon



Troisième problème: On a perdu toute notion de densité des POI. L'Espagne n'a pas l'air tellement moins couverte que le centre de la France.





Autre artefact amusant: des POI clusters dans l'eau.



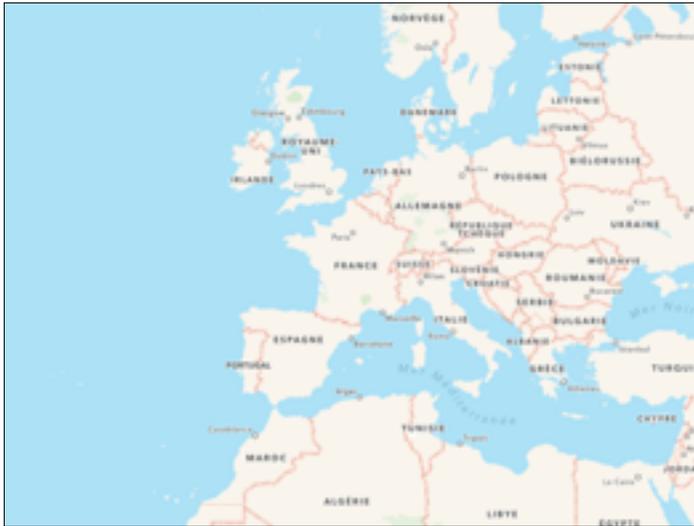
Une variante de l'algorithme de clustering par quadtree. Le but est de prendre en compte la densité des POI. Ça marche pas mal, même un peu trop: on a perdu les points isolés.

Autour de Paris, on a une bonne idée de la densité des gares.

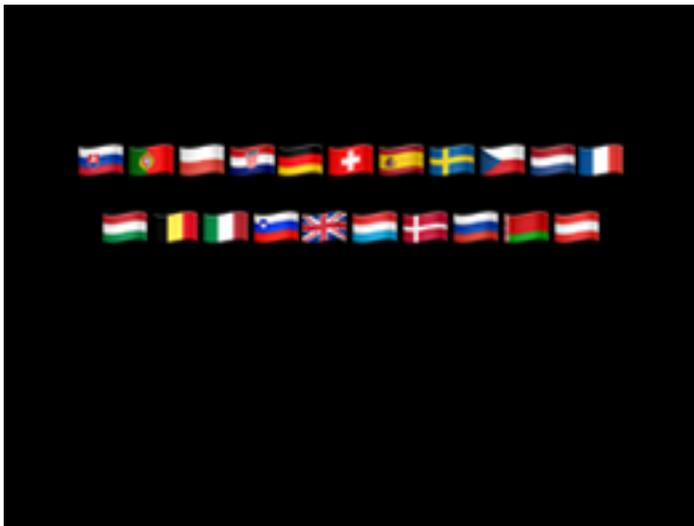


Mais en fait, là aussi on a des artefacts bizarres: un des choix est d'afficher un nombre constants d'annotations sur la carte. Quand on se déplace vers une zone vide, des clusters apparaissent et disparaissent. Et là aussi, on a des clusters dans l'eau.





Le problème fondamental, c'est que ces clusters n'existent pas: ils ne représentent aucun objet, aucune entité dans votre modèle de donnée. Ils ne sont que des groupes d'objets, plats. Les vrais POIs ont une vraie structure; c'est du modèle de donnée que doit venir la notion de groupe, la hiérarchisation.

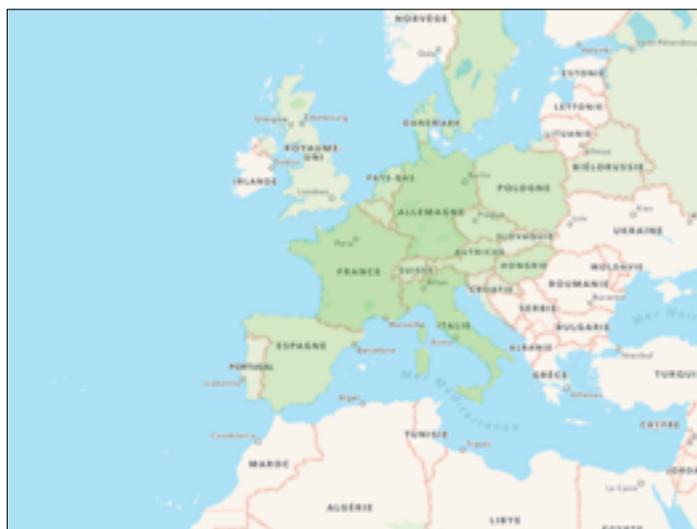


En fait, il y a une astuce facile, qui fonctionne même si les POIs n'ont pas de structure naturelle: on peut grouper par Pays (région/département/...)



Concrètement, pour les stations, ça veut dire dessiner les pays en overlay.

(Pour MapKit, c'est MKOverlay. C'est facile.)

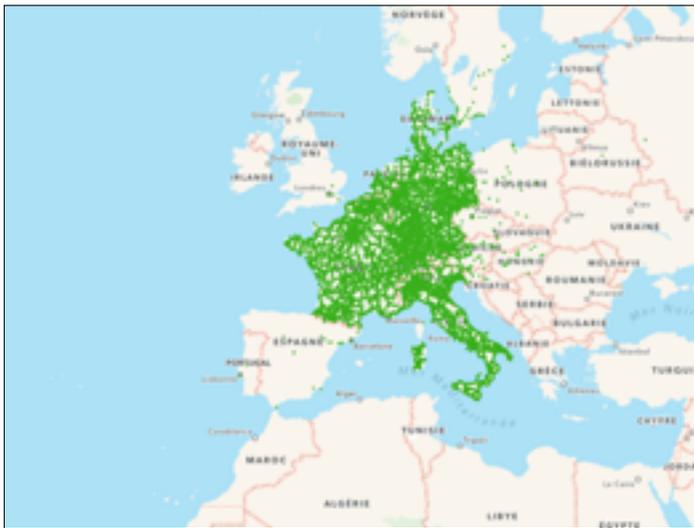


Et on peut afficher des informations statistiques directement sur les overlay. Ici, l'opacité du vert dépend du nombre de gares dans le pays.

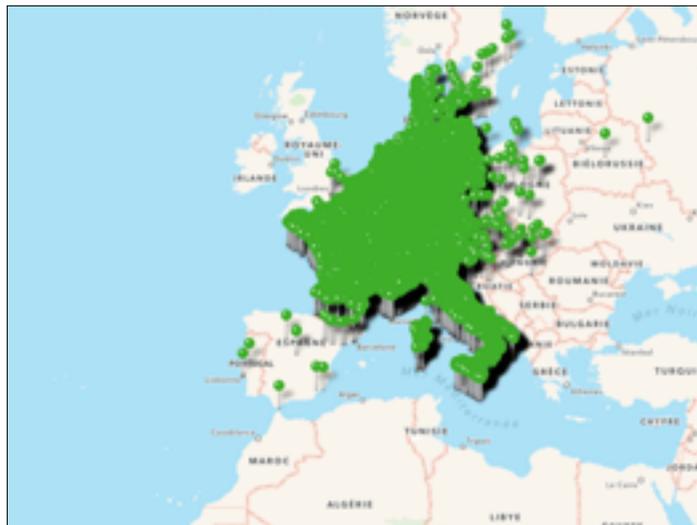


Il n'y a pas d'interaction possible dans MapKit avec les overlays, mais on peut très bien mettre une annotation pour interagir directement avec le groupe.

Ça marche plutôt bien pour afficher des informations statistiques à très haut niveau. Mais il va bien falloir afficher les points en se rapprochant.



Pour un niveau de détail intermédiaire, au lieu d'afficher chaque station comme annotation, on peut les dessiner toutes en overlay. Pas d'interaction possible avec chaque POI, mais on voit se dessiner les détails.



Qu'on ne voyait pas tout à l'heure à cause des punaises.



En zoomant, on révèle plus de détails. On voit la structure du réseau ferré!

Et en se rapprochant encore, on commence à distinguer les points individuels.



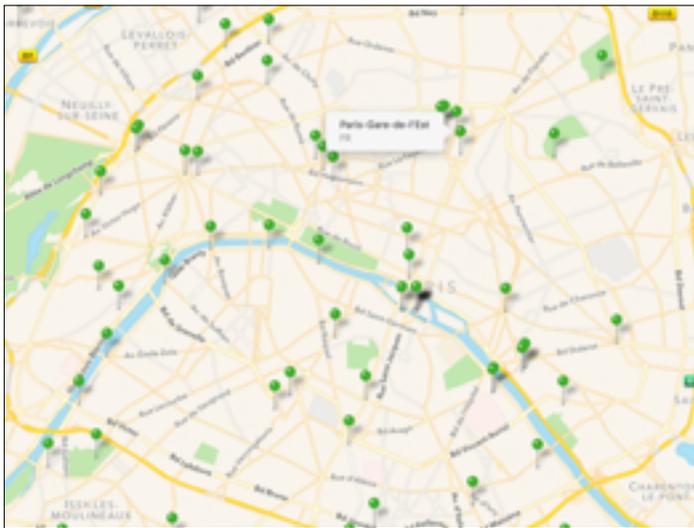
On a donc ces trois niveaux de détail:

- \* au niveau macro, des groupes avec des informations résumées,





\* au niveau intermédiaire, un overlay de points individuels, mais sans interaction,



\* au niveau le plus détaillé, chaque point est une annotation que l'utilisateur peut sélectionner.

**Merci.**

Remarque: une carte est ce qu'on appelle une ZUI, pour Zooming User Interface, un concept d'interface où les utilisateurs zooment pour entrer dans les groupes, au lieu d'ouvrir des dossiers dans des fenêtres.



## Liens

<https://github.com/n-b/MapClusters>  
<https://github.com/captaintrain/stations>

<https://github.com/choefe/CCHMapClusterController>  
[http://applidium.com/news/trop\\_de\\_pictos\\_sur\\_votre\\_carte/](http://applidium.com/news/trop_de_pictos_sur_votre_carte/)

[https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)  
<https://en.wikipedia.org/wiki/Quadtree>

<https://github.com/n-b/MapClusters>  
<https://github.com/captaintrain/stations>

<https://github.com/choefe/CCHMapClusterController>  
[http://applidium.com/news/trop\\_de\\_pictos\\_sur\\_votre\\_carte/](http://applidium.com/news/trop_de_pictos_sur_votre_carte/)

[https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)  
<https://en.wikipedia.org/wiki/Quadtree>

## Plus de liens

[https://prezi.com/qncp58\\_wfidt/clustering-of-geo-pois/](https://prezi.com/qncp58_wfidt/clustering-of-geo-pois/)  
<http://www.technology-ebay.de/the-teams/ebay-kleinanzeigen/blog/ios-mapkit-clustering.html>  
<https://robots.thoughtbot.com/how-to-handle-large-amounts-of-data-on-maps>  
<https://infinum.co/the-capsized-eight/articles/a-blazingly-fast-open-source-algorithm-for-poi-clustering-on-ios>

[https://en.wikipedia.org/wiki/Zooming\\_user\\_interface](https://en.wikipedia.org/wiki/Zooming_user_interface)  
<https://en.wikipedia.org/wiki/Archy>

[https://prezi.com/qncp58\\_wfidt/clustering-of-geo-pois/](https://prezi.com/qncp58_wfidt/clustering-of-geo-pois/)  
<http://www.technology-ebay.de/the-teams/ebay-kleinanzeigen/blog/ios-mapkit-clustering.html>  
<https://robots.thoughtbot.com/how-to-handle-large-amounts-of-data-on-maps>  
<https://infinum.co/the-capsized-eight/articles/a-blazingly-fast-open-source-algorithm-for-poi-clustering-on-ios>

[https://en.wikipedia.org/wiki/Zooming\\_user\\_interface](https://en.wikipedia.org/wiki/Zooming_user_interface)  
<https://en.wikipedia.org/wiki/Archy>