

État de l'art et suggestions pour la cartographie des données acoustiques mobiles

SAMUEL MERMET ET ARTHUR DUJARDIN

École Nationale des Sciences Géographiques, France

L'équipe *Noise-Planet* (IFSTTAR/CNRS) a développé une application pour smartphone, *NoiseCapture*, afin de recueillir des données collaboratives sur le niveau sonore dans l'environnement, dans un objectif de produire des cartes de bruit. Les données sont accessibles via une plateforme web, mais l'évaluation de la qualité de ces données, ainsi que leur représentation, est en cours de réflexion. Le présent article présente des solutions quant à la correction des données géographiques associées aux données acoustiques. Différentes pistes de cartographie du bruit, intégrant un indice de confiance sur la qualité de la donnée, sont également proposées.

Catégories et description du sujet: Géomatique, Ingénierie

Mots clés: Cartographie, Représentations de bruits, Incertitude, Visualisation de données, OpenStreetMap, Map Matching

1. INTRODUCTION

En zone urbaine ou industrielle, le bruit impacte la qualité de vie et la santé des populations. Les cartographies stratégiques, qui permettent d'identifier les zones à enjeux en croisant niveaux de bruit et dénombrement de la population impactée, sont généralement réalisées à partir de modèles, basés sur des sources identifiées. Le bruit routier est ainsi généré à partir de données de trafic préalablement calculées, couplées à des phénomènes physiques ayant un impact sur la propagation du son. Une fois ces modèles contrôlés localement, ils sont extrapolés sur des territoires plus vastes. Le bruit ambiant ainsi obtenu diverge parfois du bruit pouvant être ressenti par les habitants. L'application mobile Android *NoiseCapture* développée par l'équipe *Noise-Planet* permet de réaliser une mesure collaborative du bruit (Picaut et al., 2019). Au fil de la croissance de la communauté d'utilisateurs, la densité d'observations recueillies dans certaines zones, en Europe et dans le monde, permet de produire des cartes de bruit environnemental au moins aussi fines que les modèles traditionnellement utilisés. La base de données collectées à travers le globe est disponible en open data, permettant des exploitations variées à un public ayant un

regard suffisamment critique sur les résultats fournis. En effet, si l'application est interconnectée avec un ensemble de géoservices web qui permettent de collecter, analyser, partager et représenter les mesures collectées, ce dispositif n'offre actuellement aucun prétraitement sur les données mises à disposition, hormis une agrégation des données sous forme d'hexagones.

Dans ce cadre, notre article propose d'explorer les différentes questions nécessaires à l'analyse et à la représentation de ces données collaboratives géolocalisées aux qualités hétérogènes. Ces questions abordent différents aspects de l'analyse géomatique, de la constitution de filtres sur la qualité liée à la localisation de la donnée au choix d'agrégation adéquat pour rendre les résultats intelligibles à des décideurs, non nécessairement experts en données acoustiques.

Ainsi les données mobiles récupérées sont soumises à de nombreux aléas, intrinsèques ou non à l'appareil. Notre étude souhaite identifier les différents capteurs gérés par Android pouvant être sollicités pour définir ces filtres. Outre les valeurs fournies par le système d'exploitation, nous souhaitons les interroger à l'échelle de la trace, mais également de son contexte géographique.

En outre, nous nous intéresserons aux corrections qu'il est possible d'appliquer à ces données, notamment en s'appuyant sur des données externes telles qu'un réseau piétonnier.

Une fois ces données filtrées et corrigées, elles doivent être agrégées pour constituer un jeu de données comparable dans le temps et l'espace. Les avantages et inconvénients des modes de représentation sélectionnés pourront être discutés en matière de nombre de mesures et d'homogénéité des données collectées, mais également de leur compatibilité avec la représentation de plusieurs niveaux de lecture.

Notre troisième partie sera l'occasion de présenter et d'appliquer un algorithme de correction des traces géographiques applicable aux données de *NoiseCapture*. Le choix de cet algorithme, qui permet de rendre cohérent une trace GPS avec un réseau piétonnier ou routier préexistant, repose sur l'hypothèse que les piétons se déplacent majoritairement le long des routes et chemins.

2. QUALITÉ ET CORRECTION DES DONNÉES COLLECTÉES

2.1. Qualité géographique de la géolocalisation

Chaque modèle de téléphone possède une précision GNSS (Géolocalisation et Navigation par un Système de Satellites) indiquée par le fournisseur, soumise à de fortes altérations selon la réception du signal dont bénéficie le capteur, comme l'attestent de nombreuses études, en milieu urbain (Merry and Bettinger, 2019) ou forestier (Tomastik et al., 2016).

2.1.1. Position

La littérature indique que les erreurs de localisation sont plus fréquentes au début de la trace, lorsque l'application est lancée sans bénéficier d'une couverture satellite suffisante. Cela se traduit par des points accumulés à un même endroit, jusqu'à ce que le signal se raccroche à trois satellites minimum (El-Rabbany, 2006).

Certains décrochages satellitaires peuvent subvenir pendant l'acquisition, lorsque la couverture GNSS n'est plus assurée. Les données *NoiseCapture* intègrent une précision de localisation (*accuracy*) renseignée par le système Android. D'après la documentation du système d'exploitation (<https://bit.ly/3aN2FH4>), cette valeur, renseignée en mètre, correspond à un rayon définissant une aire circulaire autour de la localisation estimée, au sein de laquelle Android estime que la localisation réelle se situe avec une probabilité de 68%. Toutefois, dans les canyons urbains, cette valeur est susceptible d'être largement surestimée pour ce qui concerne les GPS, alors qu'elle est légèrement sous-estimée pour la localisation utilisant les ondes Wifi (Park et al., 2014). Une étude approfondie et

actualisée sur le sujet pourrait inviter les programmeurs d'application à favoriser la localisation par Wifi que celle par GPS lorsque les valeurs de précision sont suffisamment proche, afin de limiter les erreurs induites par le manque de fiabilité de cet indicateur pour le GPS.

Lorsque les coordonnées GNSS sont erronées, trop imprécises voire inexistantes, elles peuvent être approximées à partir d'un algorithme de recherche de plus court chemin ou en les combinant à l'accéléromètre du mobile (Sasiadek and Hartana, 2004), afin de donner une forme cohérente de la trace. Cette dernière technique est très répandue pour déterminer un chemin dans des conditions défavorables ou en intérieur. Cependant plus ce délai est long, plus il sera difficile de déterminer cette section manquante. Dans ce cas, la trace peut être divisée, et chaque partie peut être analysée individuellement.

Dans le cadre de *NoiseCapture*, un point important concerne la distinction entre relevés effectués en extérieur, permettant d'alimenter des indicateurs de bruit environnemental, et relevés en intérieur, susceptibles de couvrir d'autres usages. Les approches les plus efficaces pour détecter depuis un téléphone le fait qu'il se situe en intérieur ou en extérieur peuvent être regroupées en deux catégories (Bisio et al., 2015). La première regroupe les détecteurs s'appuyant sur les capteurs de l'appareil (luminosité, interface radio, magnétisme, ultrasons), la seconde sur la baisse de signal. Dans les deux cas, ces protocoles ont été mis en place lors de l'acquisition et ne peuvent être appliqués aux données disponibles a posteriori, telles que celles utilisées lors de cette étude. Si *NoiseCapture* propose déjà à l'utilisateur d'ajouter manuellement une étiquette intérieur/extérieur (entre autres choix), une mise à jour enregistrant et analysant les valeurs enregistrées par ces capteurs supplémentaires permettrait de proposer de façon automatique cette étiquette, au prix, toutefois, d'une consommation plus importante de la batterie.

2.1.2. Vitesse

La vitesse de déplacement de l'individu peut également servir comme indicateur de qualité de la donnée. Cette vitesse peut être récupérée directement grâce à l'accéléromètre du mobile (*speed*) puis analysée en étudiant les dynamiques de marche à pied ou de course pour obtenir des résultats plus fins (Cavagna et al., 1976), ou calculée manuellement grâce aux positions GNSS. Cette dernière méthode est moins fiable que la première, notamment lorsque l'environnement ne permet pas une bonne couverture et réception GNSS, mais elle permet de générer une valeur lorsque celle de l'accéléromètre n'est pas fournie.

Cette vitesse peut également indiquer si l'individu utilise un autre moyen de transports que la marche à pied (vélo, bus, voiture, etc). D'après Su et al. (2015), les méthodes de classification basées sur les données GPS

faisant appel au Support Vector Machine (SVM) sont parmi les plus populaires. D'autres méthodes utilisent des algorithmes et données basés sur les Systèmes d'Information Géographique (SIG). Xiao et al. (2015) ont élaboré une méthode s'appuyant sur un réseau bayésien pour associer à chaque vitesse, et en fonction des variations d'orientation, une probabilité qu'elle réfère à un mode de transport. Les très bons résultats produits par cette dernière étude, et la simplicité des capteurs d'ors-et-déjà enregistrés par l'application *NoiseCapture* nous invitent à préconiser cette méthode afin de générer une étiquette sur le mode de transport utilisé.

2.2. Correction des traces collectées : Map Matching

Dans un contexte où le réseau piétonnier est suffisamment bien renseigné, nous pouvons poser l'hypothèse que les contributeurs de *NoiseCapture* se déplacent en grande majorité sur des chemins identifiés. Le réseau évoqué ici est celui d'*Open Street Map* et les algorithmes identifiés pour mettre en cohérence les traces obtenues via *NoiseCapture* avec les tronçons du réseau sont ceux de *Map Matching* (Newson and Krumm, 2009) et filtres (Martin et al., 2006).

Le principe du *Map Matching* consiste à faire coïncider une trace GNSS à un réseau d'un Système d'Information Géographique. Ce problème peut être divisé en deux catégories : *online* et *offline Map Matching*. Le premier vise à corriger et recalculer en temps réel les coordonnées GNSS de l'appareil sur le graphe, utilisé dans les systèmes de navigations. Le second vise à corriger la trace une fois que l'acquisition est terminée. Pour une cartographie des données sonores mobiles, le cas *offline* est donc plus adapté.

Ces méthodes reposent sur la théorie des chaînes de Markov, l'algorithme de Viterbi (fig. 2) étant le plus répandu pour ce type d'application.

Plusieurs indicateurs peuvent qualifier ce rabattage de la trace, tel que la distance moyenne de la projection (à comparer à la précision indiquée par le téléphone), le nombre de points non reliés (le graphe routier n'est pas forcément connexe).

3. VISUALISATION

3.1. Cartographies sonores

3.1.1. Cartes de chaleur et cartes ponctuelles

Une des applications des données collectées par *NoiseCapture* est d'enregistrer une densité suffisante de mesures acoustiques pour permettre l'élaboration de cartographies du bruit. Ces dernières, faisant l'objet d'une directive "Bruit environnemental" (<https://bit.ly/2GeQRiN>), sont aujourd'hui généralement issues de modèles, basés sur des sources identifiées telles que le bruit routier. Elles sont généralement représentées sous forme de cartes de chaleur. Un premier frein à l'utilisation de ce mode de représentation est l'absence de couverture homogène de la donnée *NoiseCapture*. L'absence de donnée ne signifiant pas l'absence de bruit, l'interpolation utilisée par les cartes de bruit risque de générer des valeurs fausses entre deux mesures. De plus, nous n'avons pas identifié de mode de représentation de l'incertitude qui soit compatible avec ce type de cartes.

En dehors de ces cartes de bruit officielles, nous pouvons nous faire une idée de la diversité des cartes sonores en explorant le recensement effectué par Pierre Aumond sur son site *Sound Cartography* (<https://soundcartography.wordpress.com/>). Ces dernières sont majoritairement ponctuelles, parfois de façon regroupées, permettant d'écouter un extrait sonore attaché à un lieu. Ce mode de représentation ne correspond pas au projet *NoiseCapture*, qui s'attache à la reproduction d'un niveau sonore de l'environnement à travers une multitude de captures.

De façon plus originale, nous pouvons également citer la carte synesthésique développée sous forme de plugin QGIS (Abdelberly et al., 2014) qui génère un son en fonction des éléments observés sur une carte topographique.

3.1.2. Agrégation maillée

Une première méthode compatible avec les données issues de *NoiseCapture* consiste à mailler la surface de la Terre, et d'agréger les données correspondant à une même maille. Cette méthode est actuellement utilisée par le biais d'une grille hexagonale, permettant de représenter de manière plus fine des déplacements transversaux, en comparaison à un quadrillage. On notera que la projection cartographique utilisée par défaut par *Leaflet*, la librairie *javascript* exploitée par *Noise-Planet*, est le Web Mercator. Comme toute projection conforme, cette représentation est soumise à des distorsions de distances et d'aires, ici particulièrement importantes au niveau des pôles.

Pour se soustraire des déformations dues à la projection, une solution est d'utiliser une sphère géodésique basée sur des hexagones et des pentagones (Sahr et al., 2003). Ce principe a été développé en C++ avec le projet H3 (<https://eng.uber.com/h3/>) de la compagnie Uber. Un module *Python* est également proposé, mais sa prise en main est complexe et moins bien documentée.

L'équipe de *Noise-Planet* limite la visualisation de ses résultats acoustiques à des hexagones de 15m de côté, distance sur laquelle ils considèrent que le son varie peu. Ils génèrent des mailles de tailles supérieures, mais elles se limitent à afficher le nombre de mesure.

3.1.3. Agrégation au tronçon routier

Les traces collectées par les applications, notamment de course à pied, sont souvent représentées par une couleur vive semi-transparente sur un fond noir. La superposition de ces traces informe visuellement sur la densité plus ou moins fortes de ces traces en un même lieu. A partir des traces reprojctées sur les tronçons piétonniers, vues au paragraphe dédié au *map-matching*, nous pouvons obtenir une agrégation des données au niveau de chaque tronçon piétonnier et, par jointure attributaire, routier. Elle permet ainsi de synthétiser une autre information que la seule densité de mesures, en l'occurrence le niveau sonore et l'indice de confiance qui est attribué à l'information affichée. Un exemple de cartographie présentant un résultat visuellement similaire est la *Chatty Map* (Aiello et al., 2016), associant à chaque tronçon routier une catégorie (transport, nature, human, music, building) en fonction de données issues des réseaux sociaux.

La première limite de cette méthode est qu'elle nécessite un réseau dense. Pour y remédier, une densification du réseau peut s'effectuer à partir d'une généralisation des traces des usagers (Kasemsuppakorn and Karimi, 2013). La seconde limite identifiée concerne l'échelle de représentation des tronçons routiers. En effet, plus on réduit cette échelle plus on obtient un "effet spaghetti" rendant complexe la compréhension de la donnée. Plusieurs études se sont intéressées à la simplification automatique de lignes (Shi and Cheung, 2006), qui peuvent être appliquées aux traces GNSS. Ces méthodes peuvent être améliorées en fixant des contraintes spatio-temporelles (Qian and Lu, 2017).

3.2. Représentation d'un indice de confiance

La première partie de cet article a listé les différents points pouvant être pris en compte pour mettre en place un protocole de nettoyage ou de qualification des données. Une réflexion similaire pourrait porter sur la constitution d'un indice de confiance sur les données acoustiques, pour lequel nous renvoyons le lecteur à Botteldooren et al. (2015). A défaut d'aboutir à un indicateur à l'issue de cette première partie, nous avons généré des données factices afin d'explorer les possibilités de représentation de l'incertitude.

Boukhelifa et al. (2012) ont listé les variables visuelles pouvant être sollicitées pour représenter l'incertitude, en les regroupant en trois catégories. Les techniques basées sur les couleurs tout d'abord, en particulier la teinte que Xie et al. (2006) considère comme variable possédant la meilleure capacité à transmettre des informations de qualité, devant la luminosité. Elle peut être couplée à un ajustement de la largeur de la ligne. La saturation, mais également le blanchiment, peuvent également transmettre

une notion de dégradation de la qualité de façon intuitive, mais difficile à lire en détail. Dans ces différents cas, il faut se restreindre à un jeu de couleur unifié, cette variable visuelle n'étant pas ordonnée. Ensuite, les techniques basées sur la mise au point jouent sur la netteté des contours, la transparence et la résolution. Si le flou associe une idée d'ambiguïté aux données, il n'est pas adapté aux données quantitatives. Enfin, les techniques basées sur la géométrie sont applicables aux lignes. En particulier, le rendu non photoréaliste (NPR) correspond à l'utilisation de traits en esquisse. Le nombre croissant de perturbations dans la géométrie correspond à un plus fort niveau d'incertitude. Pour les données surfaciques, il est plus efficace de s'appuyer sur le style de bordure, comme étudié par Kübler et al. (2019).

3.2.1. Hexagones

Dans le cadre de *NoiseCapture*, nous traduisons cette hiérarchie par l'utilisation d'une couleur unique, dont la teinte varie du clair vers le foncé en fonction de l'indice de confiance apportée aux données.

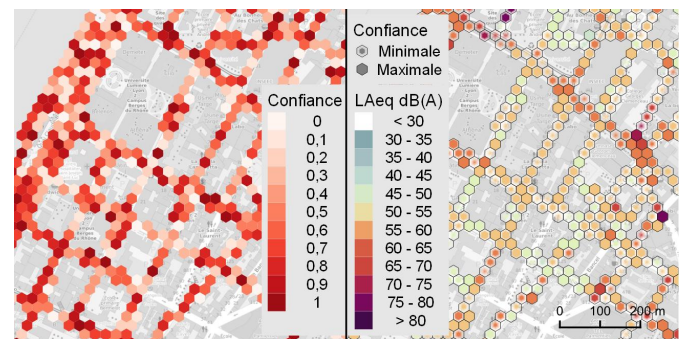


Figure 1: Représentations de l'indice de confiance (factice). A gauche il constitue la variable principale, à droite la variable secondaire après le niveau sonore.

Toutefois, cette solution n'est pas compatible avec l'utilisation d'un autre indicateur, en l'occurrence le niveau sonore moyen. Ce dernier, correspondant au premier niveau de lecture, utilise les variables visuelles de couleur et de teinte (Tomio, 2017). Nous ne pouvons pas non plus faire appel au blanchiment, étant donné que le blanc est utilisé dans le schéma de couleur de *NoiseCapture*. Pour permettre un deuxième niveau de lecture indiquant le niveau de confiance, notre proposition est d'appliquer un effet de flou aux extrémités de l'aplatissement de couleur. Ce flou est proportionnel à la confiance liée aux données : plus cet indice sera élevé, plus l'hexagone sera rempli. Les bords de l'hexagone, non concernés par ce flou, permettent de rappeler la taille maximale de l'entité.

3.2.2. Tronçons

L'efficacité, potentielle, de la proposition précédente repose sur l'homogénéité de l'information spatiale surfacique portant l'information. Ces deux aspects ne sont pas applicables à la représentation des données sous forme de tronçons. À défaut d'une solution faisant l'unanimité dans la littérature, nous préconiserions, à l'instar de Boukhelifa et al. (2012) et Limberger et al. (2016), l'utilisation de traits en esquisses pour représenter l'incertitude sur une donnée. La mise en place, associant une plus forte variabilité dans le tracé selon le degré d'incertitude, est détaillée par Martín et al. (2019).

Ces deux propositions doivent faire l'objet d'un protocole de validation que nous n'avons pas pu mettre en place dans la durée de cette étude.

4. APPLICATION : CORRECTION DES TRACES GNSS PAR MAP MATCHING

4.1. Données utilisées

4.1.1. Choix de la zone d'étude

Nous avons travaillé sur les données collectées sur la métropole de Lyon entre septembre 2017 et avril 2018, du fait de la densité de données collectées par l'application *NoiseCapture*. Ainsi, 500 traces ont été récupérées, pour un total de 399852 points.

4.1.2. Données *NoiseCapture*

Les données utilisées sont les traces brutes, directement téléchargées sur le serveur via un script Python. Les fichiers obtenus sont au format *geojson* pour les traces, et *properties* pour les meta données.

4.1.3. Données *OpenStreetMap*

Le réseau *OpenStreetMap* est un graphe orienté, contenant à la fois les axes principaux routiers mais également secondaires, telle que des chemins ou trottoirs. Un axe est formé par plusieurs noeuds et côtés, accessible via un identifiant. Ce sont ces derniers objets élémentaires qui ont été utilisés comme base pour l'algorithme de map matching.

4.2. Algorithme de Map Matching

Pour notre étude, nous nous sommes intéressés uniquement au développement d'algorithmes de *Map Matching* de type *offline*. Les traces étant récupérées une fois le trajet fini, nous avons trouvé ce choix plus pertinent. Pour chacun des algorithmes, la trace est projetée en coordonnées Web Mercator. Le calcul des distances euclidiennes ou des longueurs de chemins sont fait en prenant l'ellipsoïde *WGS84* comme référence.

4.2.1. Algorithme de Viterbi

L'algorithme de *Viterbi* repose sur une chaîne de *Markov* cachée, et fait un compromis entre rabattre la trace sur le tronçon le plus proche, et le tronçon le plus cohérent. Dans ce modèle, chaque point de la trace mesurée constituent les états observables de la chaîne de *Markov*, tandis que les points corrigés (projetés sur le tronçon routier) constituent les états cachés. La suite des états cachés la plus probable (la route suivie sur le réseau) est déterminée en comparant la distance euclidienne de l'état caché à l'état observable et du chemin routier entre deux états cachés successifs. Newson and Krumm (2009) utilisent la fraction, Lou et al. (2009) la différence, et Wei et al. (2012) ne tiennent compte que de la longueur du chemin.

4.3. Développement effectué

Afin de mesurer l'apport de l'algorithme de *Viterbi*, nous avons développé un algorithme de rabattage orthogonal de la trace au tronçon routier le plus proche, que nous qualifierons de "naïf" par la suite. Les traces corrigées ainsi que des indicateurs statistiques ont été stockées dans une base de données SQL, afin de comparer et d'évaluer les performances du modèle.

Le code est en open source sur *github* <https://www.github.com/arthurjdj/noiseplanet>, ainsi que sa documentation sur <https://noiseplanet.readthedocs.io>. Plusieurs bibliothèques en dépendent, telle que *numpy*, *osmnx*, *json*, *pandas*, *pyproj*. Concernant l'algorithme de *Viterbi*, le package provenant de l'université de Leuven, disponible lui aussi en open source sur *github*, a été utilisé. Néanmoins, les fonctionnalités ont été modifiées pour répondre précisément aux besoins de l'équipe *NoiseCapture* et être compatibles avec leurs données.

4.4. Résultats

4.4.1. Améliorations vis-à-vis de l'algorithme du plus proche voisin

L'algorithme de *Map Matching* sur le tronçon le plus proche présente de nombreux défauts. Le premier est que le chemin généré sur le réseau routier peut ne pas suivre la trace mesurée (fig. 2). De plus, le graphe *OpenStreetMap* n'étant pas connexe, certaines points de la trace peuvent être projetés dans des cours intérieurs ou espaces cloisonnés. De ce fait, le chemin le long du réseau sera disjoint.

L'algorithme basé sur les chaînes de *Markov* remédie à ce problème, et la trace corrigée est continue dans le réseau. Cependant, la distance de la projection est logiquement plus élevée que dans l'algorithme naïf et l'écart entre la trace originale et résultante est plus importante (fig. 2).

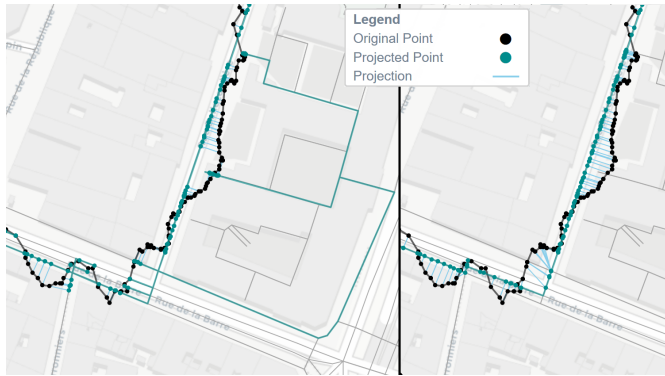


Figure 2: Comparaison des résultats de Map Matching sur une même trace par la projection orthogonale (à gauche) et l'algorithme de *Viterbi*. Le trajet noir correspond au chemin libre (non rabattu sur le réseau *OpenStreetMap*). Le trajet bleu clair correspond au chemin reliant les points corrigés sur le réseau routier le plus fin. On note que l'algorithme naïf présente un chemin moins cohérent que dans la seconde illustration. En effet, projeter la trace sur le tronçon le plus proche n'implique pas forcément de minimiser le chemin parcouru. Ainsi, il se peut que deux points consécutifs soient projetés sur deux voies routières proches, mais pas reliées directement. Le chemin reliant ces deux points (calculé via l'algorithme de *Dijkstra*) présentera donc un détour.

4.4.2. Performances

Nous avons conçu trois indicateurs pour comparer les résultats du rabattage orthogonal et de l'algorithme de *Viterbi*. La longueur de la projection *proj long*, entre le point fourni par Android et sa correction, est logiquement à l'avantage de l'algorithme naïf, sans que la différence des moyennes à l'échelle des près de 400,000 points relevés ne soit exagérément prononcée (7.1m contre 8.6m). Nous avons de plus contextualisé ces valeurs en les comparant avec la précision planimétrique indiquée par Android, en gardant à l'esprit l'imprécision de cette information en milieu urbain. Ainsi, 65,4% des points projetés orthogonalement se situent dans le disque dont le rayon est défini par la précision, contre 55,7% pour les projections utilisant les chaînes de Markov. Ces chiffres pourront être considérablement améliorés en retirant ou recalculant les points trop éloignés du réseau piétonnier.

A contrario, l'algorithme de *Viterbi* assure la continuité de la trace sur le réseau. Dans un cas sur mille, l'algorithme naïf projette en effet les points sur des parties du graphe piétonnier *OpenStreetMap* qui ne sont pas connectées entre elles (ou via un parcours trop long). Plus précisément, nous avons calculé la longueur de trajet *chemin* entre deux points projetés consécutifs.

Les moyennes obtenues sont respectivement de 9.3m et de 6.8m pour les algorithmes naïf et de *Viterbi*.

4.4.3. Visualisation des résultats

Les points corrigés par algorithme de Map Matching, hébergé sur un serveur de l'Unité Mixte de Recherche Acoustique de l'Environnement, ont ainsi pu être agrégés, tant sous la forme d'hexagones que de tronçons routiers. Dans le second cas, l'ensemble des données sont bien agrégées au réseau routier et non piétonnier afin d'améliorer la lecture du résultat et la robustesse de la mesure.

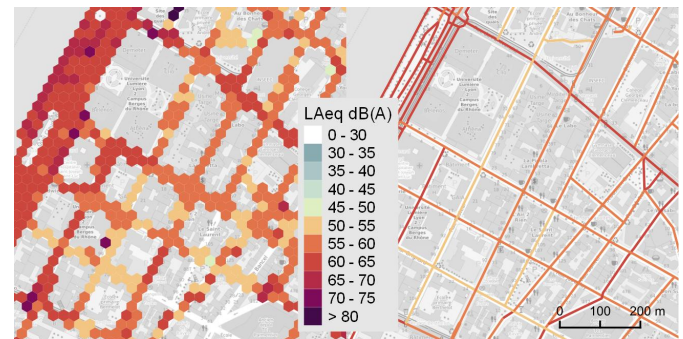


Figure 3: Comparaison des données agrégées par hexagones et par tronçons.

Pour l'agrégation de données reprojetées sur le réseau routier *OpenStreetMap*, l'intérêt des hexagones est limité à la précision de son maillage. L'exploitation de nos données génère ainsi 18,448 hexagones contre 4,024 tronçons. Toutefois, cette précision géographique limite de fait le nombre de mesures collectées, et donc la robustesse de l'indicateur statistique retenu. En effet, le nombre de point moyen (médian) par tronçon est bien supérieur à celui par hexagones : 99.4 points en moyenne (21 en médiane) par tronçon contre 21.7 (8) par hexagone, sur une des zones les mieux fournies en mesures *NoiseCapture* en France. La différence entre le nombre de traces représentées est plus limitée : respectivement 2.6 et 2.2 traces par tronçon et hexagones, pour une même médiane, égale à 1. Ces différences de taille d'échantillon ont également un impact sur les indicateurs de dispersion appliqués aux mesures acoustiques. En effet, l'écart-type moyen est plus élevé à l'échelle des tronçons (4.5dB) qu'à celle des hexagones (3.8dB). Les conclusions apportées par ces indicateurs simples sont limitées par le manque de filtres préalables aux données et par la comparaison d'objets géographiques différents. Si nous n'avons pas exploré cette piste d'analyse statistique, il est possible de travailler sur des unités spatiales comparables en associant une surface

aux tronçons routiers en multipliant leur longueur par leur largeur (définie par défaut si non renseignée).

Un autre avantage de la donnée linéaire est de pouvoir joindre une seconde variable visuelle, l'épaisseur, pour retranscrire des informations secondaires, notamment le nombre d'observations.



Figure 4: Représentation du niveau sonore par tronçon et du nombre d'enregistrements associés.

5. CONCLUSION ET PERSPECTIVES

À défaut de répondre à l'ensemble des questions soulevées en introduction, nos premiers résultats tendent à valider l'utilisation d'un algorithme de Map Matching, en l'occurrence celui basé sur les chaînes de Markov, pour corriger les traces GNSS des données *NoiseCapture*. Le développement d'un package python facilitera l'intégration de cet outil à la chaîne de traitement de *Noise-Planet*, apportant à brève échéance une amélioration des coordonnées spatiales. Pour de meilleures performances, cet algorithme devra intégrer une zone tampon afin d'éviter des projections trop lointaines. Du fait de sa dépendance au réseau piétonnier fourni, nous avons également préconisé la lecture de travaux permettant de compléter un graphe piétonnier à partir de traces GNSS collaboratives, telles que celles fournies par *NoiseCapture*.

De plus, nous avons identifié d'autres pistes d'amélioration du dispositif. Si notre recherche ne nous a pas permis d'identifier de solutions justifiées pour filtrer les données de localisation aberrantes, il est possible d'apporter des éléments de contexte, notamment sur le mode de transport utilisé, à partir des données actuellement collectées par *NoiseCapture*. La détection de positionnement en intérieur ou en extérieur est également bien documentée, mais les propositions retenues demandent une mise à jour de l'application afin d'intégrer des capteurs supplémentaires.

La solution d'agrégation proposée actuellement par l'équipe de *Noise-Planet* a pu être confrontée à une alternative offrant des combinaisons de variables visuelles

plus lisibles et une généralisation aux échelles inférieures mieux documentée. Les avantages et inconvénients de ces deux niveaux d'agrégations doivent toutefois encore être développés au niveau statistique. Enfin, l'efficacité des propositions intégrant des représentations croisées devra être mesurée, notamment par une enquête.

RÉFÉRENCES

- Abdelbery, S., Bonin, O., Josselin, D., and Altman, E. (2014). Propositions pour une cartographie sonore, synesthésique et interactive. In *Spatial Analysis and GEOmatics (SAGEO 2014)*, Grenoble, France.
- Aiello, L. M., Schifanella, R., Quercia, D., and Aletta, F. (2016). Chatty maps: constructing sound maps of urban areas from social media data. *Royal Society open science*, 3(3).
- Bisio, I., Delfino, A., and Lavagetto, F. (2015). A simple ultrasonic indoor/outdoor detector for mobile devices. In *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 137–141.
- Botteldooren, D., Can, A., and Michel, S. (2015). Projet grafic, cartographie fine et continue de la qualité sonore des lieux et des parcours dans l'espace urbain. rapport intermédiaire.
- Boukhelifa, N., Bezerianos, A., Isenberg, T., and Fekete, J.-D. (2012). Evaluating sketchiness as a visual variable for the depiction of qualitative uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2769–2778.
- Cavagna, G., Thys, H., and Zamboni, A. (1976). The sources of external work in level walking and running. *The Journal of physiology*, 262:639–57.
- El-Rabbany, A. (2006). *Introduction to GPS: The Global Positioning System*. Artech House Publishers; 2nd Revised edition.
- Kasemsuppakorn, P. and Karimi, H. (2013). A pedestrian network construction algorithm based on multiple gps traces. *Transportation Research Part C: Emerging Technologies*, 26:285–300.
- Kübler, I., Richter, K.-F., and Fabrikant, S. I. (2019). Against all odds: Multicriteria decision making with hazard prediction maps depicting uncertainty. *Annals of the American Association of Geographers*, 0(0):1–23.
- Limberger, D., Fielder, C., Hahn, S., Trapp, M., and Döllner, J. (2016). Evaluation of sketchiness as a visual variable for 2.5d treemaps.
- Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., and Huang, Y. (2009). Map-matching for low-sampling-rate gps trajectories. pages 352–361.

- Martin, J., Krösche, J., and Boll, S. (2006). Dynamic gps-position correction for mobile pedestrian navigation and orientation.
- Martín, D., Arroyo, G., Sol, V. d., Romo, C., and Isenberg, T. (2019). Analysis of drawing characteristics for reproducing traditional hand-made stippling. 80:1 – 16.
- Merry, K. and Bettinger, P. (2019). Smartphone gps accuracy study in an urban environment. *PLoS ONE*, 14:e0219890.
- Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. pages 336–343.
- Park, M., Kim, H., Lee, S., and Bae, K. (2014). Performance evaluation of android location service at the urban canyon. In *16th International Conference on Advanced Communication Technology*, pages 662–665. IEEE.
- Picaud, J., Fortin, N., Bocher, E., Petit, G., Aumond, P., and Guillaume, G. (2019). An open-science crowdsourcing approach for producing community noise maps using smartphones. *Building and Environment*, 148:20 – 33.
- Qian, H. and Lu, Y. (2017). Simplifying gps trajectory data with enhanced spatial-temporal constraints. *ISPRS International Journal of Geo-Information*, 6:329.
- Sahr, K., White, D., and Kimerling, A. J. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2):121–134.
- Sasiadek, J. and Hartana, P. (2004). Gps/ins sensor fusion for accurate positioning and navigation based on kalman filtering. *IFAC Proceedings Volumes*, 37:115–120.
- Shi, W. and Cheung, C. (2006). Performance evaluation of line simplification algorithms for vector generalization. *The Cartographic Journal*, 43:27–44.
- Su, H., Caceres, H., Tong, H., and He, Q. (2015). Travel mode identification with smartphones. *Sensors*, 15:16.
- Tomastik, J., Sr, J., Salon, S., and Piroh, R. (2016). Horizontal accuracy and applicability of smartphone gnss positioning in forests. *Forestry*, 90:187–198.
- Tomio, B. (2017). Cartographic visualisation of noise and aspects of public understanding of this information. In *Participatory Sensing, Opinions and Collective Awareness*, pages 181–199. Springer.
- Wei, H., Wang, Y., Forman, G., Zhu, Y., and Guan, H. (2012). Fast viterbi map matching with tunable weight functions. pages 613–616.
- Xiao, G., Juan, Z., and Zhang, C. (2015). Travel mode detection based on gps track data and bayesian networks. *Computers, Environment and Urban Systems*, 54:14–22.
- Xie, Z., Huang, S., Ward, M. O., and Rundensteiner, E. A. (2006). Exploratory visualization of multivariate data with variable quality. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 183–190.