Construction de la cartographie d'une plateforme aéroportuaire à partir de traces avions

Cindie Andrieu [1] & Zakaria Bezza [1]

[1] *Safety Line, 130 rue de Lourmel, 75015 Paris, France*
cindie.andrieu@safety-line.fr, zakaria.bezza@safety-line.fr

Résumé. Le développement des processus d'acquisition des traces numériques a entraîné récemment un véritable essor des méthodes de construction automatique de carte, notamment dans le domaine routier. L'intérêt de ces méthodes est d'une part, de ne pas dépendre d'un format propriétaire de cartographie, et d'autre part, de disposer d'une cartographie à jour. Dans cette étude, nous proposons d'appliquer ces méthodes, utilisés essentiellement dans le domaine routier, à l'analyse du trafic sur une plateforme aéroportuaire. Nous proposons ainsi de reconstruire la cartographie d'un aéroport sous la forme d'un graphe représentant l'ensemble des voies de circulation à partir des traces radar des aéronefs ayant circulé sur la plateforme.

Mots-clés. Algorithmes de construction automatique de carte, traces numériques.

Abstract. Due to the availability of tracking data with the development of acquisition techniques, map construction methods have been developed considerably, especially in road transport. The advantages of these methods is on the one hand, not to depend on a proprietary format, and on the other hand, to have an up-to-date map. The aim of this study is to apply these methods, mainly used in the road sector, to the trafic analysis on an airport platform. We propose to generate the map of an airport, represented as a graph that corresponds to taxiways, from the radar traces of the aircraft which move on the platform.

Keywords. Map construction algorithms, tracking data.

1 Introduction

Afin d'analyser le trafic sur une plateforme aéroportuaire, il est nécessaire de disposer d'une carte de l'ensemble des voies de circulation des avions et des véhicules. Plusieurs choix sont alors possibles : obtenir une carte préexistante sous un format propriétaire (ex : NavTeq, TeleAtlas, fichiers AutoCAD propres au gestionnaire d'aéroport), ou utiliser une base de données géographiques libre telle que OpenStreetMap [5]. Cependant, le premier choix rend difficile l'extrapolation des résultats de l'analyse à n'importe quel aéroport par la dépendance d'un format spécifique qui peut différer d'un aéroport à l'autre. Quant

au deuxième choix, il ne permet pas de disposer d'une cartographie à jour incluant les modifications d'infrastructures ou de procédures.
Afin de pallier ces inconvénients, les algorithmes de construction automatique de carte à partir de traces numériques ont connu récemment un véritable essor, notamment dans le domaine routier avec le développement des processus d'acquisition des traces GPS (ex : smartphones). Ces méthodes appelées « map construction algorithms » consistent à reconstruire la cartographie d'un réseau routier sous forme d'un graphe (chaque arête correspondant à un segment du réseau), à partir des trajectoires des mobiles circulant sur ce réseau.
L'objectif de cet article est d'appliquer les méthodes de construction automatique de cartes, utilisés essentiellement dans le domaine routier, à l'analyse du trafic sur une plateforme aéroportuaire. Nous proposons ainsi de reconstruire la cartographie d'un aéroport sous la forme d'un graphe représentant l'ensemble des voies de circulation à partir des traces radar des aéronefs ayant circulé sur la plateforme. Après une présentation rapide des différents algorithmes existants, nous présentons une application à la construction du graphe de la plateforme Roissy-Charles-de-Gaulle (CDG).

2 Les algorithmes de construction automatique de carte

Le développement des processus d'acquisition des traces numériques a entraîné récemment un véritable essor des méthodes de construction automatique de réseau routier à partir notamment de données GPS. Parmi les différents articles proposant un état de l'art de ces méthodes, on peut citer notamment Biagioni et Eriksson (2012) [2] et Ahmed et al. (2015) [1], qui distinguent trois principales méthodes :

- les méthodes de type k-means, dont le principe consiste à réduire l'ensemble des points GPS à un ensemble plus petit constitué de centroïdes, puis à relier ces centroïdes entre eux afin de former la géométrie de la route.
- les méthodes de type « trace merging », dont le principe consiste à compléter le réseau de manière récursive en fusionnant directement les nouvelles trajectoires avec le graphe préexistant.
- les méthodes de type KDE (« Kernel Density Estimation ») dont le principe consiste à construire un estimateur à noyau de la densité de traces numériques sur le réseau, puis en extraire la géométrie de la route par des méthodes de traitement d'image.

Les deux articles cités montrent que les algorithmes de type KDE tels que ceux de Davies et al. (2006) [4] et de Biagioni et Eriksson (2012) [3] sont les plus performants en terme de précision de la carte reconstruite et produisent les cartes les moins complexes (peu de noeuds et d'arêtes). Pour plus d'informations, la plupart des algorithmes récents de construction automatique de cartes, ainsi que les codes associés et différents jeux de

données, sont disponibles sur le site : `http://www.mapconstruction.org/`.

3 Descriptif de l'algorithme de Biagioni et Eriksson

L'algorithme de type KDE proposé par Biagioni et Eriksson (2012) [3] est basé essentiellement sur des méthodes de traitement d'image (le code Python est disponible à l'adresse suivante : `https://www.cs.uic.edu/bin/view/Bits/Software`).

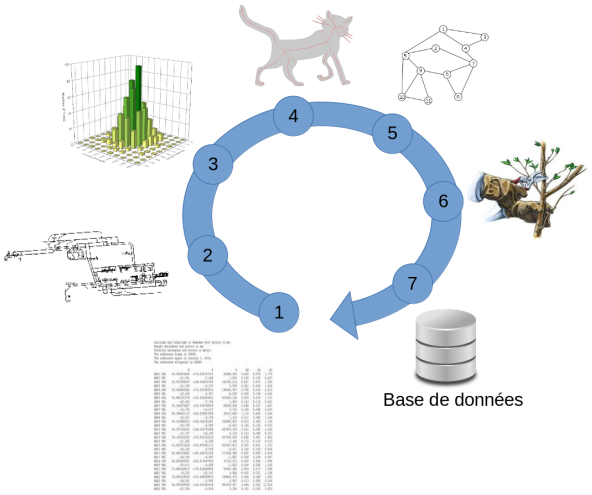

FIGURE 1 – Schéma des différentes étapes de l'algorithme de Biagioni et Eriksson.

Il se décompose en différentes étapes illustrées à la figure 1 :

1. Fichiers contenant l'ensemble des trajectoires observées sur le réseau.
2. Image de l'ensemble des trajectoires.
3. Estimation à noyau de la densité (KDE, « Kernel Density Estimation »).
   La zone d'étude est discrétisée en un maillage, et le nombre de trajectoires passant dans chaque cellule est comptabilisé afin d'obtenir un histogramme 2D. Puis, afin de pallier les erreurs de positionnement et les mauvaises identifications de routes parcourues, cet histogramme est convolé avec un noyau gaussien (filtre flou gaussien). À l'issue de cette étape, on obtient une image en niveau de gris où plus la fréquence de passage est élevée, plus le trait est sombre.
4. Extraction du squelette de l'image.
   Cette étape a pour but d'extraire le « squelette » de l'image obtenu à l'étape précédente en utilisant l'algorithme du « skeleton » proposé par Zhang et Suen (1984) [8]. Cependant, afin de tenir compte de la densité de fréquentation des segments et du bruit de mesure des données de position, Biagioni et Eriksson

ont proposé d'étendre l'algorithme du « skeleton » afin d'obtenir un squelette de l'image en niveau de gris (« gray-scale skeleton ») plutôt qu'une image binaire.

5. Génération du graphe par l'algorithme de « Combustion ».
   Cette troisième étape consiste à générer un premier graphe à partir du squelette de l'image obtenu à l'étape précédente. Dans un premier temps, les intersections de route sont détectées à partir du squelette de l'image en utilisant l'algorithme de « combustion » décrit dans Shi et al. (2009) [6], puis des noeuds conservant la géométrie de la route sont ajoutés au graphe. Certains noeuds intermédiaires ne modifiant pas la géométrie de la route sont supprimés par l'utilisation de l'algorithme de Douglas-Peucker.
6. Élagage du graphe via une étape de map-matching basée sur l'algorithme de Viterbi.
   Cette étape consiste à projeter l'ensemble des trajectoires originales sur le graphe obtenu à l'étape précédente via l'utilisation d'un algorithme de map-matching afin de supprimer les arêtes peu vraisemblables. La méthode de map-matching utilisée par Biagioni et Eriksson est basée sur l'algorithme utilisé dans le système VTrack proposé par Thiagarajan et al. (2009) [7]. L'algorithme VTrack utilise l'algorithme de Viterbi où le problème de map-matching est modélisé dans un contexte de modèle de Markov caché (HMM, « Hidden Markov Model »). L'objectif est alors, pour l'ensemble des points d'une trajectoire, de trouver la séquence de segments parcourus la plus probable sur le réseau routier modélisé par le graphe.
7. Sauvegarde du graphe sous forme d'une base de données.
   A l'issue de l'algorithme, on obtient un graphe orienté constitué de deux tables de données :
   - « nodes » : table des noeuds, chacun étant caractérisé par un identifiant et ses coordonnées géographiques (latitude, longitude) ;
   - « edges » : table des arêtes où chaque arête relie deux noeuds et est caractérisée par un identifiant, les identifiants des noeuds de départ et d'arrivée de l'arête, et un poids correspondant au nombre de trajectoires "bien" map-matchées sur celle-ci.

4 Application à la construction du graphe des voies de circulations des avions d'une plateforme aéroportuaire

Nous proposons d'appliquer l'algorithme de Biagioni et Eriksson présenté à la section précédente, à la reconstruction du graphe des voies de circulation des avions de la plateforme Roissy Charles-de-Gaulle (CDG). Pour cela, nous avons sélectionné un ensemble de 14289 traces avions (cf. figure 2) représentatives de l'ensemble des voies de circulation

des avions, des différentes configurations de la plateforme (face à l'Est ou face à l'Ouest pour CDG), et des différentes actions de vol (décollage, atterrissage, roulage).

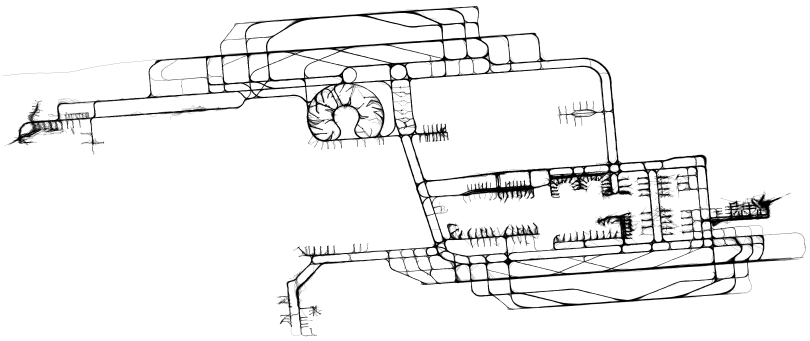

FIGURE 2 – Ensemble des traces avions sélectionnés pour la construction du graphe de CDG.

À l'issue de l'algorithme de Biagioni et Eriksson, nous avons obtenu un graphe constitué de 2186 noeuds et 4174 arêtes. Cependant, ce graphe présentait quelques problèmes de topologie (arêtes non connectées alors qu'elles devraient l'être), ainsi qu'un trop grand nombre de noeuds et d'arêtes. Afin de pallier ces inconvénients, nous avons procédé à une nouvelle étape d'élagage et de raffinement topologique afin de simplifier le graphe tout en conservant la géométrie de la route et la topologie du réseau. Le graphe obtenu est illustré à la figure 3 : celui-ci est composé de 1241 noeuds et 3128 arêtes.

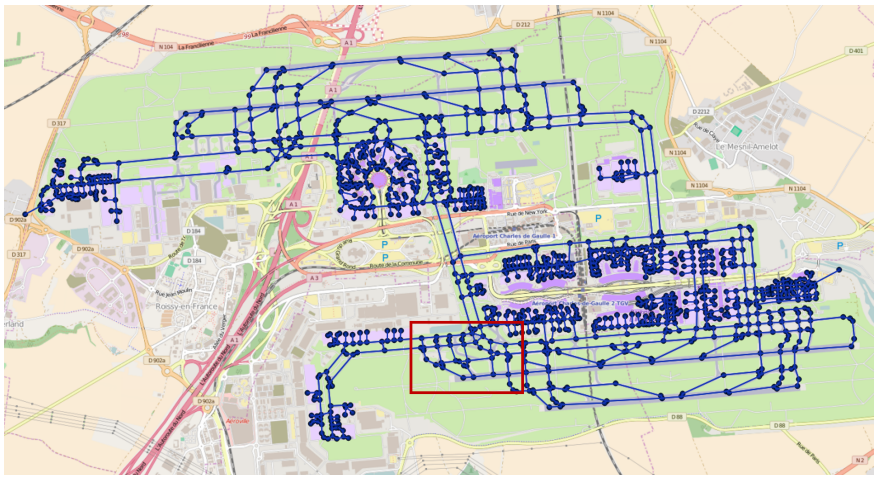

FIGURE 3 – Graphe des voies de circulation de CDG sur le fond de carte OpenStreetMap.

On observe que le graphe obtenu est bien représentatif de l'ensemble des voies de circulation des avions sur CDG même s'il reste quelques arêtes aberrantes à l'est du terminal 2,

et à l'ouest. On observe également dans la zone encadrée en rouge que la carte OpenStreetMap n'est pas à jour contrairement au graphe, des travaux ayant modifiés la disposition des taxiways.
L'intérêt de disposer d'un tel graphe est d'une part, de ne pas dépendre d'un format spécifique pouvant différer d'un aéroport à l'autre, et d'autre part, de disposer d'une cartographie à jour. D'autre part, une fois le graphe construit, il est possible de procéder à un map-matching des trajectoires, favorisant ainsi la mise en oeuvre de techniques d'apprentissage et de clustering afin d'analyser les flux de trafic sur la plateforme.

Remerciements

Les travaux présentés sont en partie issus du projet FUI « AWACS » soutenu par la BPI et le Conseil Régional d'Ile-de-France. Les auteurs remercient ces institutions pour leur soutien ainsi que l'ensemble des partenaires du projet : le Groupe ADP, l'IFSTTAR/GRETTIA et le LIP6.

Bibliographie

[1] Ahmed, M., Karagiorgou, S., Pfoser, D., and Wenk, C. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica 19*, 3 (2015), 601–632.

[2] Biagioni, J., and Eriksson, J. Inferring road maps from global positioning system traces : Survey and comparative evaluation. No. 2291, Transportation Research Board of the National Academies, pp. 61–71.

[3] Biagioni, J., and Eriksson, J. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (2012), ACM, pp. 79–88.

[4] Davies, J., Beresford, A. R., and Hopper, A. Scalable, distributed, real-time map generation. vol. 5, IEEE, pp. 47–54.

[5] OpenStreetMap. Openstreetmap. https ://www.openstreetmap.org, 2016.

[6] Shi, W., Shen, S., and Liu, Y. Automatic generation of road network map from massive gps, vehicle trajectories. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on* (2009), IEEE, pp. 1–6.

[7] Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., and Eriksson, J. Vtrack : accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (2009), ACM, pp. 85–98.

[8] Zhang, T., and Suen, C. Y. A fast parallel algorithm for thinning digital patterns. vol. 27, ACM, pp. 236–239.