



HAL
open science

Localisation indoor à l'aide des capteurs d'un smartphone

Jean-Gabriel Krieg

► **To cite this version:**

Jean-Gabriel Krieg. Localisation indoor à l'aide des capteurs d'un smartphone. Autre [cs.OH]. Institut National Polytechnique de Toulouse - INPT, 2017. Français. NNT : 2017INPT0009 . tel-04218625

HAL Id: tel-04218625

<https://theses.hal.science/tel-04218625>

Submitted on 26 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :

M. JEAN-GABRIEL KRIEG

le lundi 27 février 2017

Titre :

Localisation indoor à l'aide des capteurs d'un smartphone

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. ANDRE LUC BEYLOT

M. GENTIAN JAKLLARI

Rapporteurs :

Mme NATHALIE MITTON, INRIA VILLENEUVE D'ASCQ

M. NICOLAS MONTAVONT, TELECOM BRETAGNE CAMPUS DE BREST

Membre(s) du jury :

M. THOMAS NOEL, UNIVERSITE DE STRASBOURG, Président

M. ANDRE LUC BEYLOT, INP TOULOUSE, Membre

M. FRANCK ROUSSEAU, INP GRENOBLE, Membre

M. GENTIAN JAKLLARI, INP TOULOUSE, Membre

Mme HAKIMA CHAOUCHI, TELECOM SUD PARIS, Membre

On ne va jamais aussi loin que lorsque l'on ne sait pas où l'on va.
Christophe Colomb

REMERCIEMENTS

Je tiens à remercier tout particulièrement mon directeur de thèse, André-Luc Beylot, qui a su me donner les bonnes directions à suivre avec confiance et patience, ainsi que Gentian Jakllari mon co-encadrant en laboratoire pour son enthousiasme et sa disponibilité. Par ailleurs, je remercie mon autre co-encadrant, Hadrien Toma, qui s'est impliqué avec inspiration dans mes travaux et qui est devenu un véritable ami.

Je suis aussi très reconnaissant envers Nathalie Mitton, directeur de recherches à l'INRIA Lille-Nord Europe et Nicolas Montavont, professeur à l'IMT Atlantique - Télécom Bretagne - qui ont réalisé des rapports de thèse particulièrement complets et instructifs. Je remercie également le président de mon jury Thomas Noël, professeur des universités de l'Université de Strasbourg, ainsi que les autres membres du jury Franck Rousseau, maître de conférences à Grenoble INP-Ensimag et Hakima Chaouchi, professeur à Télécom SudParis. Leurs commentaires constructifs lors des discussions qui ont suivi la soutenance de la thèse ont ouvert des perspectives particulièrement intéressantes.

Je remercie aussi l'ensemble du personnel de l'entreprise Alsett et du laboratoire IRIT que j'ai pu côtoyer. Tout particulièrement, je tiens à souligner l'incroyable disponibilité de SAM, les aides précieuses de Manu, Riadh, Béatrice, Carlos, Katia, Julien, et la très bonne ambiance entretenue par les thésards Guillaume, JB, Farou9, Nesrine, Eric, Cédric, Yoann, Oana, Samer, Dorin, Elie, Justine, etc. Je me souviendrai toujours du fabuleux bureau F404 grâce à Emilie et Aziz.

Je ne peux pas éviter de remercier ma famille, mes parents, mes frères, mes belles-sœurs et mes nièces ainsi que ma belle-famille, qui m'ont amené à dépasser mes limites par leur soutien et leurs encouragements et qui ont participé parfois activement aux résultats de ce manuscrit. Je remercie aussi Yoan et Axelle pour les évasions footballistico-culinaires et Michel pour son amitié de chaque instant.

Pour finir, je réserve mon ultime remerciement à ma femme, Aurélie. Sa patience lors des nuits de travail, son attention de tous les jours, son écoute et son humour sont indispensables à mon efficacité et mon équilibre. Merci d'avoir toujours été là pour moi, je te dois tout.

RÉSUMÉ

La localisation par satellites et la démocratisation du GPS ont permis à la navigation de se développer et de devenir aujourd'hui un élément incontournable du quotidien. Les smartphones, qui accompagnent constamment leur détenteur, sont un élément clé de cette réussite. Par ailleurs, les technologies de l'information et de la communication (TIC) prennent aujourd'hui une place croissante dans les services proposés aux utilisateurs. Ces technologies, comme l'utilisateur lui-même, ne peuvent plus se contenter de la localisation en extérieur. La localisation à l'intérieur des bâtiments est donc un défi qui pourrait élargir la palette d'applications connectées utilisant cette information.

Les solutions existantes proposent d'utiliser les signaux terrestres pour identifier la position de l'utilisateur du smartphone. Elles nécessitent cependant une connaissance du bâtiment dans lequel il se trouve. Cette thèse s'intéresse au problème de localisation *indoor*, dans le but de fournir une solution de navigation équivalente à celle du GPS en extérieur, quel que soit le bâtiment. En particulier, nous étudions la faisabilité d'utiliser les capteurs de mouvement, intégrés dans les smartphones, qui forment une centrale inertielle potentiellement capable de suivre la trajectoire de l'utilisateur.

Nous proposons pour cela une application fondée exclusivement sur les données glanées auprès des capteurs du smartphone. Pour cela, nous contournons l'imprécision de la centrale inertielle du smartphone en étudiant la locomotion humaine et son effet sur les signaux des capteurs. Une campagne de tests nous a permis de montrer que nous obtenons une précision de l'ordre du mètre dans la plupart des bâtiments. Nous construisons ensuite une solution de navigation en recréant le plan du bâtiment à l'aide des capteurs d'ambiance embarqués dans les smartphones. Nous créons alors une représentation sous forme de graphe du plan pour proposer à l'utilisateur un chemin entre sa source et sa destination. Nous montrons enfin comment l'utilisation des capteurs des smartphones peut ouvrir d'autres voies. Notamment, nous proposons une solution de gestion de places de parking en ville fondée sur la connaissance du mode de transport de l'utilisateur.

ABSTRACT

Today, we can use our smartphones to drive from Los Angeles to New York, yet we can not use them to find our way around Charles de Gaulle Airport. GPS simply does not work indoors and if outside areas are mapped very well, the indoors of most buildings remain a mystery to our favorite navigation app.

State of the art solutions propose to use terrestrial signals to localize the smartphone user. However, they require accurate knowledge of the floor plan of the building where the user is located. This thesis advocates for a solution that provides the same user experience available outdoors, thanks to the GPS, inside any building, anywhere in the world.

We introduce an integrated solution that relies exclusively on the smartphone sensors. To accomplish this, we circumvent the imprecision of the smartphone inertial measurement unit (IMU) by studying human locomotion and its effect on sensor signals. Through experiments with a prototype implementation show that we can obtain meter-level accuracy in most indoor scenarios. We complete our navigation solution by creating the building floor plan using the ambience sensors embedded in smartphones. We create a graphical representation of the floor plan to be used by the smartphone user for navigation. Finally, we show that a complete understanding of smartphones sensors can open the door to other applications as well. In particular, we propose a solution for smart on-street parking that relies on a transportation detection module.

TABLE DES MATIÈRES

Table des figures	xiii
Liste des tableaux	xvii
Introduction	1
1 Techniques de localisation sur smartphone	7
1.1 Les systèmes de positionnement traditionnels et l' <i>indoor</i>	8
1.1.1 Systèmes de positionnement terrestres	8
1.1.2 Géolocalisation par satellites	9
1.1.3 Trilatération plutôt que triangulation	10
1.1.4 Sources d'erreur et augmentation de la précision	13
1.1.5 Réception du signal satellite en intérieur	15
1.2 Utilisation des signaux terrestres	15
1.2.1 Propagation des signaux en intérieur	16
1.2.2 Temps de trajet ou puissance reçue	18
1.2.3 Particularités et précision des signaux terrestres	20
1.3 Limiter les besoins en infrastructure	23
1.3.1 Modèle déterministe	23
1.3.2 Éviter la connaissance des points d'accès	26
1.3.3 Sans infrastructure : la solution du <i>crowdsourcing</i>	27
1.4 Empreintes du bâtiment : le pré-déploiement	27
1.4.1 Principe du <i>fingerprinting</i>	27
1.4.2 Estimation déterministe de la position	28
1.4.3 Estimation probabiliste de la position	29
1.4.4 Améliorations liées à la navigation	30
1.5 Modifications des smartphones	30
1.5.1 Pour le <i>Time of arrival</i>	30

1.5.2	Pour connaître l'angle d'arrivée du signal	31
1.6	Positionnement, navigation ou détection d'environnement	32
1.6.1	Localisation par rapport à un environnement	33
1.6.2	Principes de <i>deadreckoning</i>	33
1.6.3	Comptage de pas	34
1.6.4	En utilisant la caméra	36
1.7	Notre positionnement	37
1.7.1	Limiter les infrastructures et les modifications d'équipements . . .	37
1.7.2	Architecture d'une solution hybride	38
2	Acrux : une solution de navigation par différenciation de locomotions	43
2.1	Présentation d'Acrux	44
2.1.1	Cadre applicatif	44
2.1.2	Les capteurs de mouvement pour la navigation	44
2.1.3	L'idée d'Acrux et les défis à relever	45
2.1.4	Architecture de la solution	46
2.2	Les locomotions humaines	48
2.2.1	Particularités des locomotions humaines	49
2.2.2	Signature des locomotions	51
2.2.3	Détection de signatures	52
2.3	Vitesse et direction de déplacement	56
2.3.1	Mesure de la cadence des pas sur smartphone	57
2.3.2	Cadence, vitesse et locomotion	57
2.3.3	Direction de déplacement	59
2.4	Modèle 3D	62
2.4.1	Estimation du changement d'altitude	63
2.4.2	Détection du mode de transport vertical	64
2.4.3	Ascenseurs	65
2.4.4	Escalators	66
2.4.5	Escaliers	66
2.5	Performances d'Acrux	67
2.5.1	Performances <i>indoor</i>	68
2.5.2	Performances <i>outdoor</i>	70
2.5.3	Consommation de batterie	72
2.5.4	Conclusions et perspectives	72

3	Une solution pour créer une carte du bâtiment durant la navigation	75
3.1	Système InPreSS	76
3.1.1	Historique de la construction de plans	76
3.1.2	La navigation à l'estime pour la création de plans	77
3.1.3	Le principe d'InPreSS	78
3.2	Identifier les pièces à l'aide des capteurs d'ambiance	81
3.2.1	Séparation du bâtiment en cellules	81
3.2.2	Identification des murs à partir d'un capteur	85
3.2.3	Mise en commun de plusieurs capteurs	87
3.2.4	Performance du système de détection de pièces	89
3.3	Identification des chemins	90
3.3.1	Détection des portes	91
3.3.2	Du graphe au chemin	93
3.4	Cas réaliste : le bâtiment inconnu	94
3.4.1	Après le trajet d'un utilisateur	96
3.4.2	En regroupant les données de deux utilisateurs	96
3.4.3	Consommation énergétique	98
3.4.4	Conclusion	99
4	Extension de la détection de locomotion aux modes de transport	101
4.1	La détection du mode de transport	101
4.1.1	Les systèmes GPS	102
4.1.2	Les signaux terrestres	103
4.1.3	Les accéléromètres	103
4.1.4	Objectifs	103
4.2	Détection du mode de transport avec les capteurs du smartphone	104
4.2.1	Les signaux captés par le smartphone	104
4.2.2	Un système de classification approprié : <i>Random Forest</i>	107
4.2.3	Performances	112
4.3	SmartPark, une application concrète : les places de parking intelligentes	113
4.3.1	Les solutions existantes	114
4.3.2	Architecture de SmartPark	115
4.3.3	Vérification de la libération du véhicule	117
4.3.4	Évaluation et conclusion	119
	Conclusion	123

Bibliographie

129

TABLE DES FIGURES

1	La navigation <i>indoor</i> offre la possibilité d'amener l'utilisateur jusqu'à sa destination, et de ne pas arrêter le service à l'entrée de l'aéroport.	3
2	La localisation <i>indoor</i> permet un gain de temps important dans les grands centres commerciaux, et des publicités géolocalisées.	3
1.1	La trilatération en 2D nécessite la connaissance d'au moins trois distances par rapport à la balise.	8
1.2	Schéma de création d'une trame broadcast satellite, de Denelson83 (CC BY-SA 3.0)	10
1.3	Mesure de la distance en fonction du temps de propagation du signal satellite.	12
1.4	Avec trois satellites, on obtient deux points possibles, dont l'un - à gauche - est hors de la Terre. On en déduit la position du smartphone : en France. Avec quatre, on aurait, en théorie, obtenu un point unique.	12
1.5	Effet du multi-trajet sur la réception d'un signal en intérieur [1].	17
1.6	Propagation du signal dans une pièce et dans un étage. L'éloignement au point d'accès n'est pas directement lié à la puissance du signal reçu [2]. . .	17
1.7	Fonctionnement du <i>Time Of Arrival</i>	19
1.8	Le signal GSM est plus stable dans le temps que le signal Wi-Fi (802.11) [3]. La puissance de signal reçue est beaucoup moins variable sur plusieurs heures.	22
1.9	Courbe de perte de puissance du signal lors de la traversée d'un mur séparant deux pièces. Le point d'accès 1 est dans la pièce 1 et le 2 dans la pièce 2 [4].	25
1.10	Exemple de prise d'empreinte d'une pièce. Les points rouges correspondent à l'endroit de chaque scan. En bleu, les points d'accès captés.	28

1.11	La station S_1 se localise en mesurant la distance à l'émetteur - par exemple à l'aide de la puissance reçue -, et en mesurant l'angle Φ_1 par rapport à l'antenne émettrice [5].	32
1.12	La dérive d'intégration ne permet pas d'utiliser les données pures des accéléromètres pour suivre les mouvements du smartphone. Ici, l'utilisateur se déplace à une vitesse constante de $0,5m/s$, mais la vitesse estimée par intégration ne l'est pas du tout.	34
1.13	Mesure de l'accélération en fonction du temps : chaque pic point rouge correspond à un pas.	35
1.14	La forme de l'accélération pour un pas est commune à tous les êtres humains : elle est surtout marquée par un double pic au moment où le talon heurte le sol.	35
1.15	Les gyroscopes souffrent aussi de la dérive d'intégration lorsqu'on veut retrouver la direction de marche [6]. Lorsque l'utilisateur tourne de 90° puis retrouve sa position, la direction estimée par le gyroscope est biaisée.	37
1.16	Classification des solutions de localisation <i>indoor</i>	40
1.17	Notre proposition d'architecture hybride.	41
2.1	Architecture d'Acrux. Acrux utilise les capteurs de mouvement et un point de référence, appelé <i>fix</i> , pour calculer la position de l'utilisateur en temps réel. Pour cela, Acrux détermine la locomotion et la cadence de l'utilisateur.	47
2.2	Acrux convertit les données du référentiel du smartphone vers celui du bâtiment.	47
2.3	Signaux les plus importants dans la détection du pas humain pour un mouvement de deux pas.	50
2.4	Le signal d'accélération verticale reste similaire pour plusieurs positions de smartphone et différents utilisateurs.	51
2.5	Accélération verticale pour les six locomotions élémentaires.	52
2.6	Principaux signaux de capteurs pour deux locomotions différents.	53
2.7	Taux de détection de la locomotion en fonction de la taille de l'échantillon.	56
2.8	Le pic principal de la FFT permet de retrouver la période du signal : ici, c'est la cadence de pas.	57
2.9	Vitesse en fonction de la cadence pour les quatre locomotions élémentaires.	58
2.10	Estimation de la vitesse par Acrux lors d'un changement de locomotion.	59
2.11	Fonction de répartition de l'erreur dans l'estimation de la distance parcourue pour Acrux et une double-intégration.	59

2.12	Lacet et accélération latérale pendant un virage à gauche.	60
2.13	Trajectoires estimées par Acrux sans filtrage, avec filtrage et avec correction de biais pour un déplacement en U.	62
2.14	Altitude estimée par le baromètre pour des smartphones statiques pendant une heure.	64
2.15	Altitude estimée par le baromètre pour plusieurs smartphones : quatre statiques, un dans un ascenseur et un montant quatre étages par des escaliers.	64
2.16	Algorithme pour la détection des activités 3D.	65
2.17	Un utilisateur prend l'ascenseur et est suivi par Acrux.	66
2.18	Deux utilisateurs prennent un escalator, l'un le descendant en marchant, l'autre restant statique lors de la descente.	67
2.19	Acrux suit un utilisateur prenant un escalier en U.	67
2.20	Erreurs d'Acrux et des solutions de comptage de pas en fonction de la distance parcourue sur un trajet de plusieurs kilomètres.	69
2.21	Fonction de répartition de l'erreur de localisation sur l'ensemble des 460 trajets de 80 à 120m.	70
2.22	Acrux suit les mouvements de l'utilisateur dans un trajet <i>indoor</i> particulièrement difficile.	71
2.23	Acrux comparé au signal GPS sur un trajet classique : l'utilisateur part de sa station de métro préférée vers son lieu de travail.	72
3.1	Architecture de l'application InPreSS.	80
3.2	Interface utilisateur de l'application InPreSS.	80
3.3	Plan du bâtiment expérimental.	82
3.4	Partitionnement de Voronoi du bâtiment expérimental.	83
3.5	Niveau sonore dans le bâtiment.	84
3.6	Puissance du signal Wi-Fi reçue dans le bâtiment.	84
3.7	Cartes colorées de la luminosité dans notre bâtiment témoin pour deux niveaux de seuils différents.	85
3.8	Fonction de répartition de la luminosité dans le bâtiment témoin.	86
3.9	Fonction de répartition de la luminosité en échelle logarithmique pour l'abscisse.	87
3.10	Carte colorée de la luminosité en utilisant une échelle logarithmique.	87
3.11	Fonction d'efficacité du récepteur selon les valeurs du seuil. Le point rouge correspond à un seuil de 0,8.	90

3.12	Comparaison des plans obtenus par quatre méthodes de classification dont InPreSS.	91
3.13	Carte et graphe obtenus par InPreSS.	92
3.14	Graphe de la pièce B.	93
3.15	Carte finale obtenue par InPreSS. Les pointillés rouges représentent la localisation des “portes” estimée par l’algorithme.	93
3.16	Graphe à deux échelles. L’échelle macroscopique permet d’établir les liens entre les pièces. L’échelle microscopique permet d’identifier les chemins dans chaque pièce.	94
3.17	Plan du bâtiment-test.	95
3.18	Trajet du premier utilisateur (en rouge) et plan obtenu (en bleu).	96
3.19	Trajet du second utilisateur (en rouge) et plan obtenu (en orange).	97
3.20	Plan du bâtiment obtenu à l’aide de deux utilisateurs.	98
3.21	Trajet proposé (en bleu) par InPreSS pour rejoindre sa destination (étoile) à partir de sa position (cercle).	98
3.22	Consommation de batterie d’InPreSS au cours du temps.	99
4.1	Boxplot de l’accélération frontale moyenne pour les 9 modes de transport considérés.	105
4.2	Boxplot du niveau sonore pour les 9 modes de transport considérés.	105
4.3	Graphique des valeurs de niveau sonore et d’accélération appairés.	106
4.4	Une des branches finales d’un des arbres obtenus en utilisant l’algorithme de Random Forest	109
4.5	Importance des domaines choisis.	111
4.6	Importance des statistiques par type.	111
4.7	Le point de vue de l’utilisateur sur SmartPark.	116
4.8	Architecture logique de SmartPark.	117
4.9	Taux de faux positifs et de détection pour différents coefficients de similarité et différents seuils.	119
4.10	Pourcentage de détection du mode de transport pendant le stationnement.	120
4.11	CDF du temps de stationnement.	120
4.12	Consommation de batterie moyenne de SmartPark comparé au Wi-Fi et au GPS.	121

LISTE DES TABLEAUX

1.1	Principales sources d'erreurs des systèmes satellites [7].	14
1.2	Quelques valeurs de coefficients de transmission par des murs en différents matériaux pour deux fréquences du Wi-Fi [8].	17
1.3	Tableau comparatif des signaux disponibles pour la localisation.	23
1.4	Tableau des valeurs de C et γ recommandés par l'ITU pour 2,4 GHz.	24
2.1	Performance énergétique d'Acrux comparée à une solution classique de localisation de type Wi-Fi.	73
2.2	Durée de vie de la batterie de plusieurs équipements utilisant Acrux.	73
3.1	Distance utilisée pour chaque capteur.	88
4.1	Statistiques extraites de chaque capteur pour créer les variables du modèle.	107
4.2	Performances de Random Forest comparé aux autres méthodes de classification.	112
4.3	Matrice de confusion de notre solution de détection de mode de transport.	113
4.4	Comparaison entre notre solution et les solutions concurrentes de la littérature.	113

INTRODUCTION

Contexte

Se localiser, localiser ses pairs ou trouver son chemin sont des problèmes qui ont toujours existé. À chaque époque, les solutions de localisation ont utilisé les derniers moyens technologiques existants : la connaissance de la position des étoiles [9], les phares (maritimes depuis l'antiquité jusqu'à aujourd'hui, et aéronautiques depuis les années 1920 [10]), les sextants [11], les satellites [12]. Si, longtemps, la navigation s'est limitée à un positionnement par rapport à des points de référence connus (les villes, les étoiles, les phares), elle a ensuite cherché à atteindre une meilleure précision. Deux techniques se sont démarquées : le positionnement par triangulation, où l'utilisateur mesure les angles par rapport aux points de référence pour en déduire sa position (notamment grâce à l'invention du sextant au XVIIe siècle) ; et la navigation à l'estime, où l'utilisateur connaît sa position initiale, puis il suit ses mouvements grâce à des instruments de mesure comme les compas - méthode utilisée depuis le XVe siècle, et popularisée par Christophe Colomb lors de sa découverte des Amériques [13]. Ce n'est qu'à la fin du XXe siècle que les systèmes de trilatération (d'abord terrestres puis par satellites avec le GPS) ont pris le dessus, en raison de leur précision et de leur couverture [14].

À l'ère de l'information ubiquitaire, la troisième de l'informatique, les technologies de l'information et de la communication (TIC) prennent une place croissante dans notre vie quotidienne. De nouvelles applications nécessitant la localisation voient le jour avec cette nouvelle ère informatique, tels que le stationnement intelligent [15], la gestion de files d'attente [16], la surveillance de l'environnement, la gestion de la mobilité, les réseaux sociaux ou encore les jeux géolocalisés, récupèrent l'ensemble des informations captées par le smartphone de l'utilisateur. À la volonté de l'utilisateur de vouloir connaître sa position exacte, s'ajoute donc le besoin pour les applications d'obtenir cette localisation. Cela a particulièrement favorisé la démocratisation du GPS et sa présence dans tous les smartphones.

Cependant, alors que les systèmes satellitaires répondent à une grande partie des besoins, l'intérieur des bâtiments reste une zone blanche pour la localisation des équipements mobiles, car le signal GPS n'y est pas disponible (perte de signal).

Ce manque est particulièrement important pour les smartphones, éléments déterminants dans la croissance des services proposés aux utilisateurs. Il est le seul outil technologique que la grande majorité des utilisateurs transportent sur eux quasiment en permanence. Muni de plus en plus de capteurs, de mémoire et de capacités de calcul, il permet d'offrir des services aux utilisateurs, grâce aux informations géolocalisées remontées par le mobile.

La croissance de ces services a forcé la mise en œuvre de solutions de localisation complémentaires, notamment pour l'*indoor*, grâce aux autres capteurs du smartphone. Ils s'inspirent grandement des trois classes de solutions historiques :

- **installation de portiques** : le long d'un trajet, des portiques (RFID, infrarouge, etc.) sont placés et permettent d'identifier sa position [17], c'est l'équivalent des phares de l'antiquité ;
- **système de positionnement cellulaire** : le système satellitaire est copié en terrestre grâce au réseau cellulaire et/ou aux réseaux locaux. La position des cellules voisines et leur distance permet de connaître sa position [18, 19] ;
- **empreintes de réseaux locaux** : des balises Bluetooth ou des points d'accès Wi-Fi sont installés dans un bâtiment. Une empreinte des niveaux de signaux captés pour chaque position est prise. Il suffit alors de comparer son environnement en temps réel à cette empreinte pour estimer sa position [20, 21].

Pour des raisons d'imprécision et/ou de coût d'installation, ces solutions ont connu des difficultés d'implantation [22–24]. De nombreuses propositions ont émergé pour tenter de résoudre ces problèmes, mais aucune n'a suffisamment convaincu pour être déployée à grande échelle.

Cas d'utilisation

Pourtant les applications des smartphones attendent impatiemment pour permettre :

1. la continuité de service des applications fondées sur la localisation en intérieur (figure 1) [25] ;
2. les publicités géolocalisées en intérieur (centres commerciaux) [26] ;
3. la gestion de file d'attente en intérieur (guichets, centres publics) [16] ;
4. le positionnement et la navigation dans les grandes zones indoor (gares, aéroports (figure 1), musées, centre de congrès, bibliothèques) [22, 23] ;

5. le gain de temps pour des listes de tâches (supermarchés) (figure 2) ;
6. l'analyse des mouvements de foule [27] ;
7. l'étude de la couverture réseau *indoor* [28] ;
8. etc.

Les figures 1 et 2 présentent deux scénarios de navigation *indoor*. Dans le premier cas, l'utilisateur cherche son chemin pour prendre son avion. Il branche le GPS de son smartphone, suit l'itinéraire proposé jusqu'au parking. Au lieu de s'arrêter là, l'application se met en mode intérieur, et lui propose un chemin pour se rendre jusqu'à sa porte d'embarquement (A1). Dans le deuxième cas, l'utilisateur est pressé pour faire ses courses dans le centre commercial proche de chez lui. Il entre dans son application la liste des magasins qu'il souhaite visiter et les articles qu'il souhaite acheter. L'application lui propose le chemin le plus court pour valider l'ensemble de la liste de courses. En passant devant un des magasins, l'application lui envoie une offre promotionnelle pour l'inviter à entrer.



FIGURE 1 La navigation *indoor* offre la possibilité d'amener l'utilisateur jusqu'à sa destination, et de ne pas arrêter le service à l'entrée de l'aéroport.



FIGURE 2 La localisation *indoor* permet un gain de temps important dans les grands centres commerciaux, et des publicités géolocalisées.

Plus généralement, les besoins en localisation des applications se divisent en quatre grandes catégories.

- Le positionnement : le système cherche à connaître la position de l'utilisateur par rapport au bâtiment. Par exemple, un supermarché peut vouloir connaître la

- position de l'ensemble de ses clients. Cette information peut alors être partagée, ou non, avec l'utilisateur localisé.
- L'auto-positionnement : le système propose à l'utilisateur de déterminer lui-même sa position dans le bâtiment. Cela lui permettra par exemple de pouvoir se repérer en permanence.
 - L'environnement : le système ne cherche pas à connaître la position précise de l'utilisateur, mais son environnement, par exemple la pièce dans laquelle il se trouve. Un centre commercial peut vouloir compter le nombre d'utilisateurs dans chaque magasin, la position exacte des utilisateurs n'est pas nécessaire.
 - La navigation : au-delà du positionnement, l'utilisateur se voit proposer une trajectoire. C'est le cas dans les applications de navigation où le système peut proposer un chemin à l'utilisateur entre sa position courante et sa destination, et le suit tout le long de son trajet.

Les solutions existantes se concentrent sur les trois premiers domaines, en espérant étendre au besoin le positionnement à la navigation. En effet, en joignant plusieurs points de positionnement consécutifs, on obtient la trajectoire de l'utilisateur.

Motivations

L'imprécision des capteurs de mouvement (centrale inertielle) présents dans le smartphone [29] a longtemps retardé l'avènement de solutions de navigation à l'estime. Pourtant, on ne désire pas toujours connaître la position d'un utilisateur à un instant précis, mais plutôt le trajet qu'il a effectué. De la même façon que le GPS, qui est dans la plupart des cas utilisé comme un instrument de navigation et non de positionnement, une solution de localisation sur smartphone serait particulièrement efficace si elle était en fait une application de navigation.

Les solutions existantes sont également limitées par les contraintes qu'elles font peser sur l'utilisateur, les besoins en infrastructure ainsi que les coûts de déploiement.

Pour cette raison, l'objectif de la thèse est de proposer une solution de navigation à l'estime sur smartphone qui réponde aux besoins des applications : disponible partout, permettant la navigation, sans infrastructure, sans pré-déploiement, universelle et à bas coût, tout en limitant l'erreur de localisation à quelques mètres.

Plan du mémoire

Les travaux de cette thèse sont présentés en quatre chapitres.

Le chapitre 1 est un état de l'art des solutions de localisation et de navigation sur

smartphone. Étant donné la densité du domaine, nous avons choisi de distinguer les solutions suivant leurs caractéristiques, directement tirées des besoins et des contraintes des applications de localisation présentées précédemment. Nous nous positionnons ensuite par rapport à ces solutions et face aux exigences que nous nous sommes fixées, et nous envisageons l'architecture d'une solution hybride, combinant les avantages des différentes solutions identifiées.

Dans le chapitre 2, nous proposons Acrux, un système de navigation qui étudie et classifie le mouvement humain en intérieur, appelé aussi locomotion, calcule la fréquence de marche, et combine ces deux éléments pour déterminer le trajet d'un utilisateur et contourner l'imprécision des capteurs d'un smartphone. Nous présentons également un modèle 3D permettant la navigation dans des bâtiments comportant plusieurs étages. Une analyse poussée est effectuée à l'aide de données réelles prises dans plusieurs bâtiments.

Le chapitre 3 complète Acrux en lui ajoutant une reconstruction automatique du plan du bâtiment pendant la navigation - aussi appelée SLAM (*Simultaneous Localisation And Mapping*) [30]. Nous proposons un algorithme permettant de détecter les chemins qu'il est possible d'emprunter dans un bâtiment à l'aide des traces des utilisateurs. Nous montrons ensuite que les capteurs d'ambiance du smartphone permettent de retrouver le type de pièce dans laquelle l'utilisateur se trouve, ainsi que sa forme, grâce à un algorithme d'apprentissage et de classification.

Le chapitre 4 présente une extension de la méthode de classification des locomotions humaines à la détection du mode de transport. Nous étudions un système de classification de type Random Forest à partir des données des capteurs du smartphone pour la détection du mode de transport de son usager. Cet algorithme est testé sur de nombreux scénarios, sur plusieurs mois de mesure et dans plusieurs villes. Il est ensuite appliqué à un contexte concret : la gestion des places de parking en ville via notre proposition SmartPark.

TECHNIQUES DE LOCALISATION SUR SMARTPHONE

Dans ce chapitre, nous présentons les différentes solutions de localisation sur smartphone. Les solutions proposées par la littérature scientifique et par le monde industriel sont nombreuses et ont des objectifs très hétérogènes pour des applications variées. Nous avons donc décidé de classer ces solutions en fonction des critères suivants, caractéristiques de l'application et de l'utilisation visées :

1. la couverture souhaitée (extérieure/intérieure) ;
2. le type de signal disponible (GPS, Wi-Fi, GSM, etc.) ;
3. l'infrastructure présente ;
4. la possibilité de pré-déploiement ;
5. le type de smartphone des utilisateurs ;
6. le type de localisation : positionnement, auto-positionnement, environnement ou navigation ;
7. la précision souhaitée.

Dans chaque section, nous considérons les possibilités offertes par chacun de ces critères ainsi que ses limitations. Nous présentons une ou plusieurs solutions de la littérature qui paraissent représentatives des choix pris pour répondre à ces critères. Enfin, nous nous positionnons par rapport à l'ensemble de ces critères, nous récapitulons les solutions de localisation et proposons une architecture hybride combinant les avantages de certaines des méthodes présentées.

1.1 Les systèmes de positionnement traditionnels et l'*indoor*

Dans cette section, nous présentons les systèmes de géolocalisation les plus utilisés, les systèmes de positionnement par satellite, appelés *Global Navigation Satellite System* ou GNSS. Ces systèmes proposent à l'utilisateur de déterminer sa position au sol. Nous expliquons les difficultés de mise en œuvre et les sources d'erreurs, qui se retrouveront amplifiées dans un environnement intérieur. Nous montrerons alors qu'il est impossible des les utiliser dans un environnement fermé, en intérieur, sur un smartphone actuel.

1.1.1 Systèmes de positionnement terrestres

Historiquement, les premiers systèmes de positionnement à grande échelle sont terrestres. Un réseau de balises est installé dans la zone où l'on souhaite pouvoir se localiser. Il existe alors deux moyens de se positionner dans la zone :

1. **la trilatération ou mode circulaire** : l'équipement émet un signal vers la balise, qui le renvoie. L'équipement mesure alors le temps écoulé aller-retour : en supposant la vitesse de propagation constante et le trajet direct, il peut en déduire la distance qui le sépare de la balise. La position de l'équipement est alors située sur un cercle centré sur la balise. Comme le montre la figure 1.1, trois balises sont nécessaires pour trouver une position sur un plan - c'est l'intersection unique des trois cercles. SYLEDIS [31] [32] est un système de navigation qui fonctionne selon ce principe.

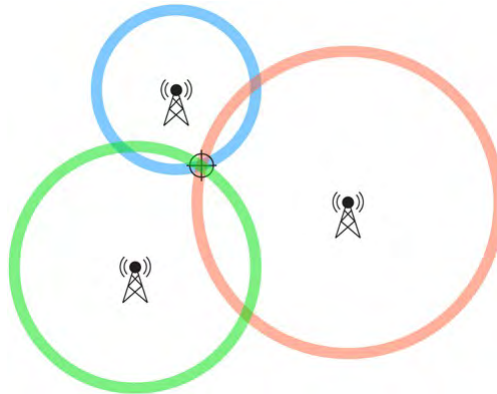


FIGURE 1.1 La trilatération en 2D nécessite la connaissance d'au moins trois distances par rapport à la balise.

2. **la différence de phase ou mode hyperbolique** : deux émetteurs émettent un signal synchronisé en phase et en temps. Le récepteur mesure la différence

entre la réception des deux signaux. Cette différence ne peut être obtenue que si l'équipement se trouve sur une hyperbole (sur une carte en deux dimensions). Pour être sûr d'obtenir un point de localisation unique, il faut trois hyperboles - l'intersection des trois hyperboles est la position -. Trois émetteurs sont donc nécessaires pour former trois paires indépendantes. DECCA [33] ou LORAN [34] sont des exemples de tels systèmes.

Ces systèmes terrestres possèdent une portée limitée, dans la mesure où ils supposent une surface de localisation plane, ce qui n'est pas le cas de la Terre. De plus, le signal envoyé est affaibli dans l'air, et peut même être dégradé par des obstacles ou par les conditions météorologiques. Malgré tout, des systèmes terrestres tels que VOR [35] ou LORAN sont toujours opérationnels, pour des applications précises comme l'aéronautique, ou en complément des systèmes satellitaires.

1.1.2 Géolocalisation par satellites

L'opportunité d'un système de localisation mondial est apparue avec la conquête spatiale dans les années 1960. Le premier système de positionnement par satellite est lancé par les États-Unis en 1964, appelé TRANSIT [36]. Il utilisait l'effet Doppler subi par le signal émis - qui contenait une localisation précise du satellite émetteur, appelée éphéméride - pour se positionner dans le référentiel terrestre géodésique.

Répondant à des besoins militaires, les États-Unis se lancent dans les années 1970 dans une solution plus précise. En effet, le système TRANSIT est limité par la propagation du signal satellite dans l'atmosphère - notamment les atténuations du signal dans la troposphère et la dispersion fréquentielle dans l'ionosphère, ce qui ne permet qu'une précision de quelques centaines de mètres. Jusqu'en février 1994, un total de trente-neuf satellites est envoyé en trois blocs pour mettre en place le nouveau système, qui a remis au goût du jour la trilatération, le GPS (Global Positioning System) [14, 37]. Dans le fonctionnement actuel, vingt-quatre satellites sont utilisés pour la localisation, quatre sont en réserve. La précision obtenue est de l'ordre de la dizaine de mètres. Pourtant, jusqu'en 2000, cette précision ne peut être obtenue que pour un usage militaire, le signal étant volontairement dégradé pour l'usage civil - on obtient alors une précision d'une centaine de mètres seulement.

Face à l'augmentation du nombre de constellations et de solutions indépendantes, l'idée de les combiner a émergé. GNSS-1 est la première génération de ces solutions combinant systèmes de positionnement par satellites, mais aussi systèmes d'augmentation de performances satellites ou terrestres pour améliorer la précision.

La deuxième génération, ou GNSS-2, inclura notamment Galileo [38, 39] et une

amélioration du GPS permettant aux deux systèmes d'obtenir une précision accrue par rapport à la première génération, mais surtout une intégrité plus importante.

1.1.3 Trilatération plutôt que triangulation

Les systèmes de positionnement par satellites sont structurés en trois composantes : le segment spatial, le segment contrôle et le segment utilisateur.

Segment Spatial

Il s'agit d'une constellation de satellites transmettant des signaux radios. Dans le cas du GPS, vingt-quatre satellites - plus certains de réserve ou de remplacement - sont disponibles plus de 95% du temps grâce à une constellation plus complète de trente-deux satellites appelée NAVSTAR. Positionnés à 20200km dans l'Orbite Terrestre Moyenne (MEO), ils suivent une trajectoire circulaire de période de révolution $11\text{h}58\text{min}$. Cette configuration permet de capter les signaux d'au moins quatre satellites en tout lieu de la surface terrestre et à tout instant.

Les satellites émettent en permanence sur deux fréquences L1 ($1575,42\text{MHz}$) et L2 ($1227,60\text{MHz}$) un message de navigation comprenant notamment leur position (ou éphémérides). Ces données sont envoyées toutes les 30 secondes dans une trame (cf figure 1.2) modulée par un code C/A (pour *Coarse Acquisition*) qui est une séquence binaire à $1,023\text{Mbit/s}$ et un code P (pour *Precision*) à $10,23\text{Mbit/s}$ chiffré et réservé aux utilisateurs autorisés par le gouvernement américain. Pour cela, les satellites sont équipés d'une horloge atomique.

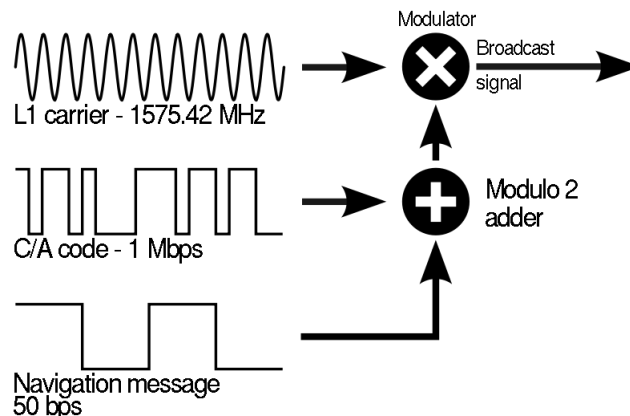


FIGURE 1.2 Schéma de création d'une trame broadcast satellite, de Denelson83 (CC BY-SA 3.0)

Le gouvernement américain se réserve la possibilité d'ajouter un code d'étalement aléatoire de précision, appelé *selective availability*, actif pour l'usage civil jusqu'en 1990, qui réduit d'un facteur dix la précision. Il leur est également possible d'interdire l'emploi du GPS dans certaines zones en chiffrant les données.

Segment Contrôle

Au sol, six stations gérées par l'armée américaine suivent les satellites, vérifient leurs transmissions, envoient des commandes de synchronisation ou de rectification d'orbite, etc. La station maître est située à Colorado Springs et fournit le temps de référence. Les autres stations sont situées non loin de l'Équateur, régulièrement positionnées autour de la Terre. Il existe également plusieurs stations de secours ainsi que des stations de suivi.

Segment Utilisateur

Il s'agit de l'équipement souhaitant se localiser, ou récepteur GPS. Il écoute les signaux et récupère les données de chaque satellite qu'il peut entendre. Il décode les données, calcule la distance qui le sépare de chaque satellite, puis estime la position de l'antenne de réception en coordonnées géographiques dans le système géodésique.

Calcul de la position

Connaissant le code C/A du signal envoyé par un satellite, le récepteur récupère la référence temporelle d'émission du signal. Il peut alors comparer l'instant d'arrivée du signal par rapport à l'instant d'émission et en déduit la distance du satellite, comme le montre la figure 1.3. Cela suppose principalement une émission en vue directe - trajet sans obstacle entre le satellite et le récepteur -, à une vitesse de propagation constante, celle de la lumière.

Avec le même principe que la trilatération présentée précédemment, chaque mesure de distance place le récepteur sur une sphère centrée sur le satellite. En théorie, il faut donc quatre satellites pour pouvoir obtenir une intersection unique. Cependant, sachant que l'intersection de trois sphères donne deux points (au plus), la plupart des récepteurs GPS prennent également en compte le fait qu'ils se trouvent forcément sur Terre pour supprimer une des deux localisations, qui apparaît comme absurde (cf figure 1.4).

En théorie, il suffit donc pour un récepteur d'avoir trois satellites en vue pour se localiser. Néanmoins, l'horloge locale du récepteur est souvent défectueuse, car il s'agit d'équipements mobiles à bas coût. Un autre satellite est donc utilisé pour corriger l'horloge du récepteur.

Comme nous l'avons vu, dans le cadre du GPS - ainsi que de la plupart des systèmes

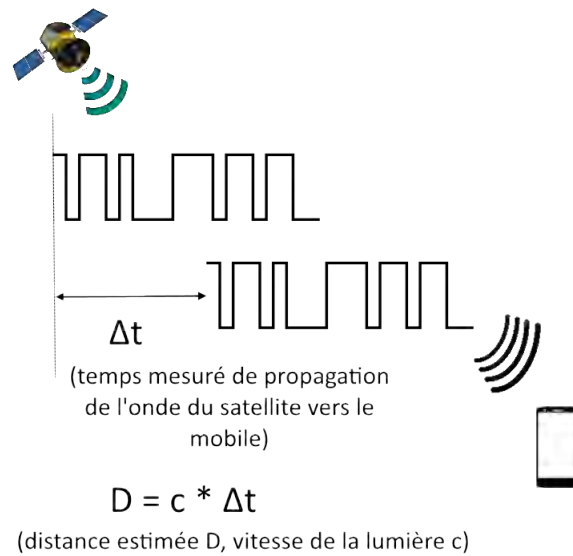


FIGURE 1.3 Mesure de la distance en fonction du temps de propagation du signal satellite.

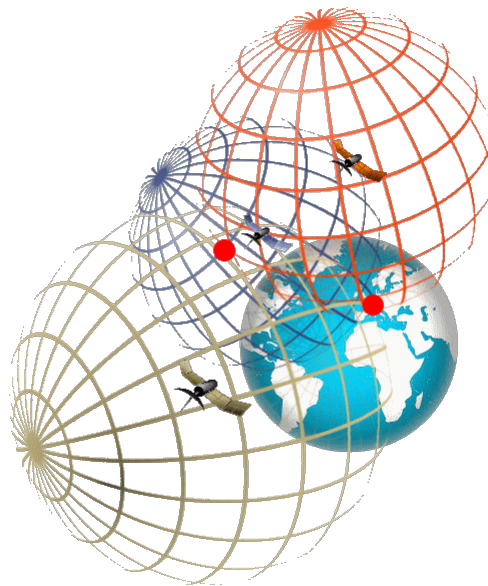


FIGURE 1.4 Avec trois satellites, on obtient deux points possibles, dont l'un - à gauche - est hors de la Terre. On en déduit la position du smartphone : en France. Avec quatre, on aurait, en théorie, obtenu un point unique.

de positionnement par satellites -, il s'agit de trilatération, et non de triangulation. Le terme triangulation, souvent utilisé à mauvais escient, est communément employé en

géométrie et en trigonométrie. La triangulation consiste à déterminer la position d'un point par rapport à des points de référence - ici, les satellites -, en utilisant les angles par rapport à ces points et non les distances. En deux dimensions, il suffit de connaître deux angles par rapport aux références pour retrouver notre position. Ce n'est pas utilisé sur les systèmes satellites, d'une part en raison de la difficulté à calculer l'angle d'arrivée d'un signal sur un système de petite taille, mais aussi parce que le trajet du signal est dévié de multiples façons lors de la traversée des différentes couches de l'atmosphère.

Système géodésique de référence : le WGS 84

Les coordonnées calculées par le récepteur GPS sont converties dans le référentiel global géodésique le plus communément utilisé : le WGS 84, mieux connu sous les termes longitude, latitude et altitude.

1.1.4 Sources d'erreur et augmentation de la précision

En théorie, la précision d'un système de positionnement par GPS est de l'ordre du mètre. En pratique, elle est à peine inférieure à la dizaine de mètres. Il existe un grand nombre de sources d'erreurs qui peuvent dégrader la précision du positionnement. Tout d'abord, les récepteurs GPS sont souvent mobiles : il existe un délai entre la première acquisition d'un signal satellite et la dernière. Cependant, ceci peut être considéré comme négligeable par rapport aux autres sources d'erreurs.

Calcul de la distance parcourue

Il existe deux sources d'erreurs liées au calcul de la distance parcourue : la première provient de l'hypothèse de trajet direct. Bien que les satellites soient quasiment à la verticale par rapport au récepteur, celui-ci peut acquérir un signal qui a subi des chemins déviés, notamment en ville. Le signal obtenu est alors la somme des signaux obtenus sur tous les chemins possibles, appelé signal multi-trajet. Cela empêche de connaître précisément l'instant d'émission. De plus, le trajet pris par le signal n'est pas forcément rectiligne ni uniforme. Le signal traverse dans un premier temps l'ionosphère, une couche ionisée par le rayonnement solaire, qui va modifier la vitesse de propagation du signal, puis la troposphère à l'humidité et à la pression variables. Le signal traverse donc l'atmosphère en subissant des modifications de vitesse et de direction de propagation. Des modèles de propagation existent et tentent de limiter ces effets. Il est également possible pour des récepteurs bi-fréquence, d'utiliser les deux fréquences d'émission du signal, qui ne seront pas affectées de la même façon pour corriger l'imprécision du calcul.

Phénomène	Écart en temps	Erreur en distance
Bruit de mesure	$\sim 1ns$	$\sim 30cm$
Incertitude sur la position et l'orbite du satellite		$\sim 2m$
Dérive d'une horloge atomique embarquée sur satellite	$\sim 5ns/jour$	$\sim 2m$
Traversée de l'atmosphère	$\sim 10ns$	$\sim 3m$

TABLE 1.1 Principales sources d'erreurs des systèmes satellites [7].

Dilution de la précision (DOP)

La distance calculée entre le récepteur et un satellite n'est donc pas précise. Par conséquent, il se peut que le point unique que l'on peut espérer obtenir par intersection n'en soit pas un. La position estimée du récepteur sera alors le point le plus probable, par exemple le barycentre des trois points d'intersection. Or, si les satellites sont proches, la précision sera atténuée, car les effets décrits précédemment s'appliquent de façon similaire aux deux signaux et les sphères de présence seront proches.

Le tableau 1.1 résume les sources d'erreurs des solutions GPS [7].

Systèmes d'optimisation

En complément des satellites positionnés en orbite moyenne, des satellites géostationnaires localisés sur une zone (SBAS - *Satellite-Based Augmentation System*) permettent d'augmenter la précision en répétant le signal, ce qui réduit la durée du trajet. Or, plus celui-ci est court, meilleure est la précision. En Europe, le service EGNOS (*European Geostationary Navigation Overlay Service*) et aux États-Unis, le service WAAS (*Wide Area Augmentation System*) proposent respectivement cinq et deux satellites en complément de la constellation GPS. Il existe également des systèmes terrestres (GBAS - *Ground-Based Augmentation System*) qui agissent de la même façon que les satellites répéteurs, mais au sol.

Navigation

Une partie des récepteurs GPS est utilisée pour la navigation, plus que pour le positionnement. Cela permet à ces récepteurs de corriger la position calculée en connaissant les positionnements précédents successifs. Des algorithmes de traitement du signal, de type filtres de Kalman, peuvent ainsi utiliser le fait, par exemple, que le mouvement d'un véhicule est régulier à petite échelle pour améliorer la localisation, éviter les sauts de

position et assurer une continuité de service.

Assistance par les pairs

Un certain nombre de recherches récentes permettent de partager les informations pour améliorer la précision de la géolocalisation par satellites. Dans [40], les auteurs proposent un réseau de smartphones Android équipés de capteurs spécifiques, partageant les informations acquises sur le canal de propagation pour améliorer considérablement la précision de localisation, jusqu'à quelques dizaines de centimètres.

1.1.5 Réception du signal satellite en intérieur

La puissance émise par le satellite est d'environ $27dBW$. Le signal parcourt ensuite plus de $20000km$. La perte en espace libre étant estimée à $182dB$ [41], nous pouvons capter, au mieux, une puissance de signal de $-155dBW$, soit $-125dBm$, ce qui est bien en-dessous du bruit ambiant (autour de $-111dBm$). Afin de parvenir à extraire le signal, les récepteurs GPS sont munis d'une antenne à fort gain avec des filtres passe-bande précis. Cela leur permet de détecter et de décoder la plupart du temps le signal, en extérieur. En revanche, la puissance du signal est encore plus dégradée dès le passage à travers un mur, jusqu'à dix à cent fois dans un bâtiment. Le GPS est donc absolument inaccessible dès l'entrée, ou même à proximité d'un bâtiment, en utilisant un smartphone classique. Les auteurs de [41] proposent une solution en ajoutant une antenne directionnelle spécifique au smartphone et en utilisant des algorithmes de traitement du signal poussés sur un serveur distant pour permettre d'acquérir une position en intérieur, avec une erreur moyenne de l'ordre de la dizaine de mètres.

Sans équipement spécifique, il n'est donc pas réaliste d'envisager d'utiliser le positionnement GPS à l'intérieur des bâtiments. Plus généralement, il est difficile de prolonger la portée des systèmes de localisation par satellites en *indoor*. Pour cette raison, un grand nombre de solutions ont proposé d'utiliser des signaux terrestres, qui subissent moins la perte de puissance lors de la traversée de l'atmosphère.

1.2 Utilisation des signaux terrestres

L'impossibilité d'acquérir un positionnement fiable en environnement intérieur via les systèmes satellitaires a fait apparaître depuis une dizaine d'années un sujet de recherche majeur : comment proposer à l'utilisateur un positionnement partout, et pas seulement en extérieur ? C'est ce que l'on appelle communément *the last-mile localization problem*

[42]. Au lieu de permettre à l'équipement de se localiser dans le référentiel terrestre WGS84, l'objectif est de le placer dans son environnement : le bâtiment. Dans notre cas, nous supposons que l'utilisateur souhaite pouvoir se localiser partout, mais aussi à chaque instant, comme indiqué dans l'introduction. Pour cette raison, nous allons nous concentrer sur les solutions qui permettent au smartphone de se localiser : c'est l'objet qui, aujourd'hui, est à (presque) tout moment à proximité de son utilisateur. Certaines solutions, ponctuelles dans le temps ou cherchant à permettre à un objet de se localiser plutôt qu'une personne, proposent des équipements spécifiques (centrale inertielle haute précision, balise fixée sur l'objet par exemple, etc.) qui n'ont pas pour vocation d'être portés à tout moment.

La majorité des solutions de localisation *indoor* proposées essaient de reproduire le GPS à l'échelle du bâtiment : les satellites sont remplacés par des points d'accès ou stations de base des réseaux environnants. Dans cette section, nous présentons les spécificités de la propagation de ces signaux en intérieur avec une analogie avec l'analyse effectuée pour le GPS. Nous dressons également un inventaire des signaux utilisables pour le positionnement *indoor* et leurs particularités. Celles-ci permettent à chaque application de choisir son propre signal, suivant ses besoins et la nature du bâtiment.

1.2.1 Propagation des signaux en intérieur

En intérieur, les signaux rencontrent un grand nombre d'obstacles différents, qui les rendent particulièrement difficiles à analyser. En effet, en plus de la perte de puissance en espace libre, le signal subit le multi-trajet, il est diffracté, subit les effets du *shadowing* et du *fading* (pertes de puissance à cause d'un obstacle), etc. [43].

Multi-trajets

En intérieur, le phénomène du multi-trajet est particulièrement important. Le signal est dévié à de multiples reprises par les murs, les obstacles, les personnes, etc. Le signal émis s'additionne donc avec lui-même, décalé dans les temps. Le signal reçu est donc extrêmement dégradé, comme le montre la figure 1.5. Il est alors difficile de connaître avec précision la puissance du signal émis, sa phase et par conséquent son "début". Cela rend impossible le recours aux solutions utilisées pour le satellite en l'état.

Atténuations

La plupart des solutions de positionnement fonctionnent avec la réception d'un signal par trajet direct. En intérieur, le trajet direct sera la plupart du temps un trajet qui

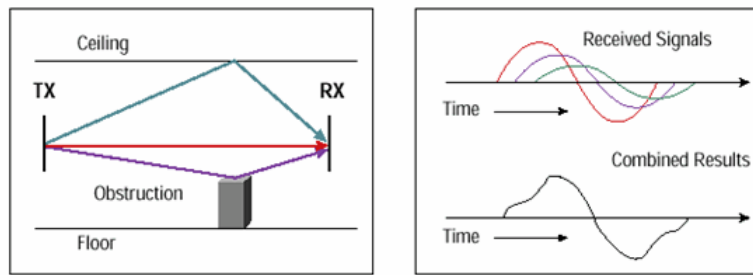


FIGURE 1.5 Effet du multi-trajet sur la réception d'un signal en intérieur [1].

Matériau	2,3GHz	5,25GHz
Plexiglass (7,1mm)	-0,3560	-0,9267
Plexiglass (2,5mm)	-0,0046	-0,2011
Verre	-0,4998	-0,0340
Cloison (12,8mm)	-0,4998	-0,5149
Briques	-4,4349	-14,621

TABLE 1.2 Quelques valeurs de coefficients de transmission par des murs en différents matériaux pour deux fréquences du Wi-Fi [8].

traverse les pièces et donc notamment les murs. Le tableau 1.2 montre que l'atténuation subie est difficile à estimer, car elle dépend de la fréquence du signal, du matériau et de la profondeur de l'obstacle. Ainsi, comme le montre la figure 1.6, il n'existe pas de relation évidente entre l'éloignement par rapport à l'émetteur et la puissance reçue.

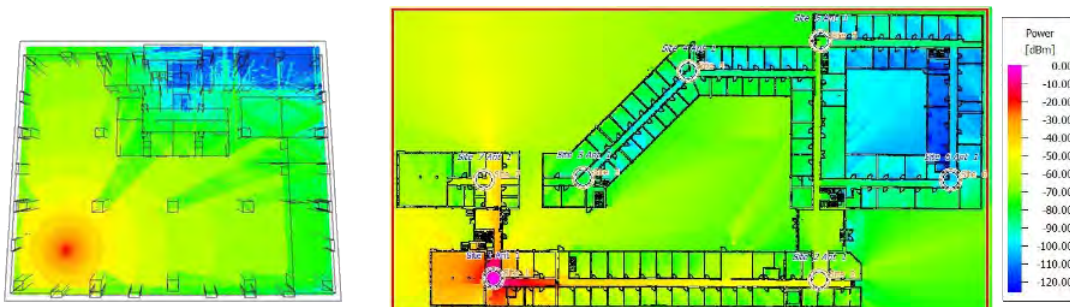


FIGURE 1.6 Propagation du signal dans une pièce et dans un étage. L'éloignement au point d'accès n'est pas directement lié à la puissance du signal reçu [2].

Diffraction et autres phénomènes

Lors de son trajet en intérieur, le signal peut également rencontrer des obstacles avec des angles spécifiques qui engendrent d'autres phénomènes. Par exemple, un signal rencontrant l'angle d'un mur sera diffracté, ce qui engendrera des interférences destructrices par endroits, avec une réception quasi-nulle du signal. L'ensemble de ces effets, connus aussi sous le nom de *fading* et *shadowing* renforce la difficulté à retrouver précisément la phase, la puissance émise et le moment d'émission d'un signal reçu en intérieur.

1.2.2 Temps de trajet ou puissance reçue

En dépit de toutes ces difficultés, des solutions sont apparues pour proposer de positionner un équipement mobile à l'aide de signaux. L'objectif est toujours de déterminer la distance du récepteur par rapport à plusieurs émetteurs, afin que le récepteur puisse retrouver sa position par trilatération. Deux grandes familles peuvent être définies : celles se fondant sur la relation puissance reçue - distance (en prenant compte uniquement l'atténuation en espace libre) et celles, plus proches du modèle satellite, calculant le temps de trajet.

Temps de trajet

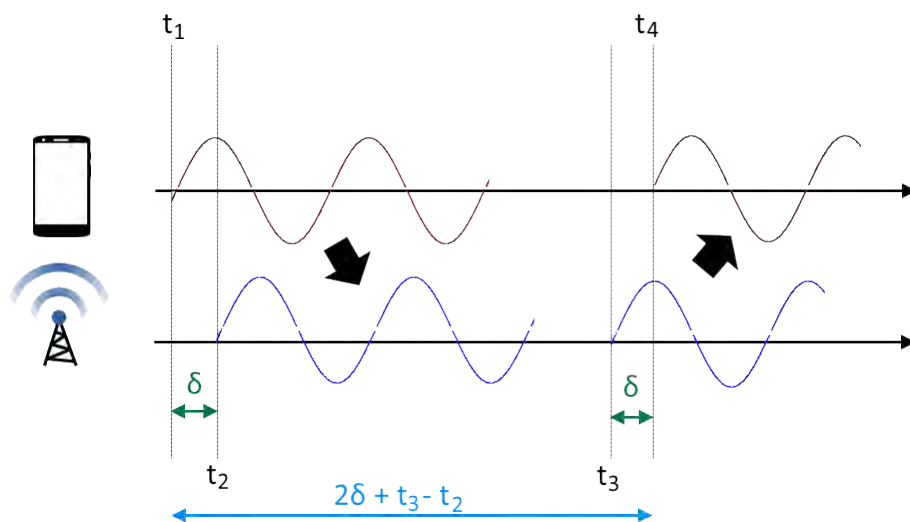
Les solutions comme PinPoint [44] qui utilisent le temps de trajet (ou TOA pour *Time of Arrival*) mesurent le temps d'aller-retour d'une communication formée d'un message et d'un accusé de réception. Un message unique ne serait pas suffisant au contraire du satellite : les points d'accès terrestres ne sont pas, dans la plupart des cas, munis d'une horloge atomique. L'aller-retour permet de s'affranchir en partie de ce besoin de précision atomique. Le fonctionnement est décrit dans la figure 1.7. À l'instant t_1 l'équipement transmet une trame. Celle-ci est reçue à l'instant t_2 par le récepteur. Au bout d'un certain temps, celui-ci va accuser réception de la trame, à t_3 . Cet accusé de réception sera reçu à t_4 . On peut alors calculer le temps de propagation δ à travers la relation :

$$t_4 - t_1 = 2\delta + t_3 - t_2. \quad (1.1)$$

Soit :

$$\delta = \frac{t_4 - t_1 + t_2 - t_3}{2}. \quad (1.2)$$

Ces solutions nécessitent une mesure précise des instants d'envoi et de réception des trames, ce qui n'est pas toujours possible avec un smartphone. Par exemple, [19, 45, 46]

FIGURE 1.7 Fonctionnement du *Time Of Arrival*.

utilisent un algorithme de TOA sur Wi-Fi. Ils supposent notamment que le temps SIFS (Short Inter-Frame Space) entre deux trames est connu ($10\mu s$ dans la norme) et espèrent que certaines trames seront envoyées directement après le SIFS. Or [47] a montré que ce délai sur des cartes Wi-Fi embarquées sur les smartphones récents est très variable, pouvant atteindre $0,8\mu s$ sur un 802.11b Samsung Broadcom. Une telle erreur sera réhibitoire dans l'estimation du temps de trajet et donc de la distance - $1,5m$ pour une erreur de seulement $10ns$ [47]. De plus, même si ces solutions fonctionnent très bien en vue directe et sont insensibles au *fading*, elles souffrent des décalages en temps engendrés par les multi-trajets. De la même façon, les solutions de type TDOA (*Time Difference of Arrival* ou *Multilateration*) subissent les mêmes effets. Ces solutions sont inspirées des premières solutions de localisation terrestre par différence de phase (mode hyperbolique) et mesurent la différence de temps de trajet entre deux émetteurs. Elles ne cherchent pas à calculer la distance exacte qui sépare l'émetteur du récepteur, mais la différence de distance entre le récepteur et les deux émetteurs.

Les solutions utilisant le temps de trajet ne sont pour le moment pas adaptées aux cartes Wi-Fi des smartphones. Cependant, elles pourraient l'être dans le futur, ou en utilisant un autre type de signal, dont on pourrait connaître l'instant exact d'émission et de réception.

Puissance du signal reçu

Les technologies utilisant la puissance du signal (ou RSS pour *Received Signal Strength*)

supposent connue la puissance d'émission. Étant donné la puissance reçue, elles calculent alors la distance grâce à la formule de perte en espace libre (FSPL pour *Free Space Path Loss*) [48] :

$$FSPL = \left(\frac{4\pi d}{\lambda} \right)^2 \quad (1.3)$$

$$= \left(\frac{4\pi df}{c} \right)^2 \quad (1.4)$$

où $FSPL$ est la perte en puissance, d est la distance en mètres, f la fréquence en hertz, λ la longueur d'onde en mètres et c la vitesse de la lumière dans l'air.

Ces solutions sont particulièrement sensibles aux différentes atténuations subies par le signal lors de sa propagation *indoor*. Elles nécessitent un traitement de signal avancé pour pouvoir fonctionner de façon précise.

1.2.3 Particularités et précision des signaux terrestres

Un récepteur terrestre mobile peut recueillir un grand nombre de types de signaux, à des fréquences différentes et en utilisant des technologies variées. Cette sous-section présente les particularités d'une partie non-exhaustive d'entre elles.

Signaux Wi-Fi

Le premier signal auquel on peut penser pour permettre à un récepteur de se localiser est le signal Wi-Fi. Il possède l'avantage de correspondre à des infrastructures existantes fortes, notamment en intérieur. Plusieurs fréquences sont utilisées, en particulier à 2,4 GHz et 5 GHz. Plutôt positionné sur de hautes fréquences parmi les signaux communément utilisés, le Wi-Fi souffre particulièrement des différentes atténuations et dégradations du signal en intérieur. RADAR [18] a été le premier papier proposant d'utiliser le signal Wi-Fi pour pouvoir retrouver la position d'un équipement en connaissant la position des points d'accès d'un bâtiment. Cette solution a également été fortement poussée par Google, qui, grâce notamment à ses reconnaissances pour son application Google Maps, a répertorié l'ensemble des points d'accès existants et a calculé leur position. Cela permet au smartphone d'obtenir assez facilement sa position grâce à son réseau Wi-Fi. La précision communément admise est de l'ordre de la dizaine de mètres, mais elle dépend énormément de la densité des points d'accès dans le bâtiment. En revanche, elle est fortement sensible aux changements de position des points d'accès et aux modifications

du bâtiment. Par ailleurs, de par la nature aléatoire de la méthode d'accès du Wi-Fi, il est aujourd'hui impossible d'utiliser une technique utilisant la mesure du temps de propagation.

Signaux téléphoniques

Les signaux téléphoniques ont également été envisagés. En effet, la couverture du réseau cellulaire est la plus importante à travers le monde. Par ailleurs, les réseaux cellulaires fonctionnent sur des bandes de fréquences réservées, ce qui les rend moins sensibles aux interférences. Enfin, le réseau cellulaire est maîtrisé par les opérateurs : on peut connaître facilement la position des stations de base qui n'est pas amenée à changer fréquemment. C'est l'inverse des solutions Wi-Fi pour lesquelles l'infrastructure et les points d'accès sont chez le client, peuvent être déplacés ou mis hors-ligne régulièrement. De part la diversité des technologies (GSM, HSPA, LTE, etc.), un grand nombre de bandes de fréquence est disponible ; elles s'étalent de 450 MHz à 3,8 GHz par exemple pour LTE. Globalement la plupart d'entre elles sont en-dessous des bandes libres utilisées par le Wi-Fi à 2,4 GHz et sont donc légèrement moins sensibles aux atténuations et aux interférences, comme le montre la figure 1.8 pour le GSM, tiré de [3]. En revanche, les stations de base sont moins denses généralement dans un bâtiment que les points d'accès Wi-Fi : il peut exister de nombreux points d'accès pour une seule station de base. Ainsi, on ne peut pas profiter de l'effet de nombre pour corriger les éventuelles erreurs. Les auteurs de [3] sont les premiers à proposer une solution précise de localisation par GSM et montrent des résultats semblables au Wi-Fi dans un bâtiment peu dense en points d'accès. Cela montre que les réseaux cellulaires peuvent être un réseau de secours dans un bâtiment où l'infrastructure Wi-Fi est insuffisante.

Bluetooth beacons

À l'opposé des caractéristiques du réseau cellulaire, beaucoup d'entreprises proposent des solutions fondées sur le *Bluetooth Low Energy*. En plus de sa faible consommation, le Bluetooth a une portée limitée. Alors que cela pourrait paraître comme un inconvénient dans une application de localisation, ces entreprises proposent de multiplier le nombre de points d'accès, ou *beacon Bluetooth*, et de profiter de sa portée limitée pour augmenter la précision. Évidemment, l'inconvénient majeur est que cela nécessite d'installer un grand nombre de beacons dans le bâtiment dans lequel on veut proposer de se positionner : au moins un par petite pièce et jusqu'à une dizaine dans les grands halls (aéroports par exemple). Insiteo [23] et Polestar [22] sont des exemples de *start-ups* françaises qui se sont lancées sur ce type de technologies. Ces solutions sont particulièrement adaptées

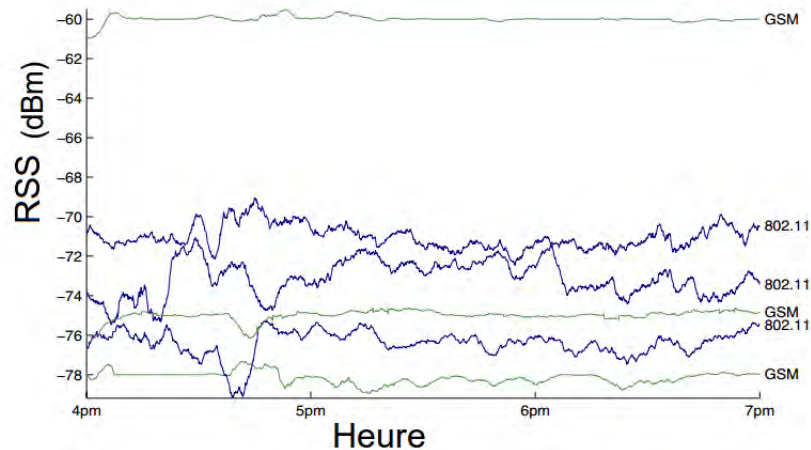


FIGURE 1.8 Le signal GSM est plus stable dans le temps que le signal Wi-Fi (802.11) [3]. La puissance de signal reçue est beaucoup moins variable sur plusieurs heures.

dans des endroits où le propriétaire du bâtiment visé peut se permettre d'installer des *beacons* un peu partout, notamment s'il a besoin d'une précision et d'une continuité de service importantes.

Signal Radio FM

Les auteurs de [49] ont proposé d'utiliser le signal radio FM. L'avantage de ce signal est qu'il utilise de plus basses fréquences que les systèmes précédemment cités. Il est donc plus résistant au *fading*, à la présence humaine et aux changements d'orientation du mobile. Malheureusement, ils ont aussi montré qu'il était presque... trop stable. Par conséquent, il est difficile de différencier la puissance reçue entre deux endroits proches, d'autant plus que les émetteurs sont en général assez éloignés. Le signal FM est donc particulièrement adapté à de grands bâtiments avec un grand nombre d'obstructions et de murs.

RFID tags, infrarouge et ultrason

Certaines solutions ont élargi l'idée d'ajout de *beacons* apportée par le *Bluetooth*. Ainsi, plutôt que d'ajouter des équipements de communication et de s'en servir comme solution de localisation, il a été proposé d'ajouter des lecteurs et émetteurs dédiés. Plusieurs technologies ont été avancées, que ce soit des tags RFID (le smartphone va scanner un code RFID pour trouver sa position) [17], des ultrasons (mesure du temps aller-retour, à la manière d'un sonar) [50] ou de l'infrarouge [51]. Ces solutions conduisent à des erreurs inférieures à celle des signaux RF classiques, mais elles nécessitent l'installation d'un grand nombre de *beacons* utilisant une technologie peu utilisée sur smartphone.

Signal	Couverture	Densité	Précision	Exemples d'application
Wi-Fi	importante en ville	très dense en ville	$\sim 8m$	localisation partout en ville
GSM	complète	assez dense en ville	$\sim 10m$	solution de réserve
Bluetooth	inexistante	nulle, ajout de <i>beacons</i>	$\sim 5m$	Congrès, supermarchés, etc.
FM	très importante	peu dense	$\sim 15m$	solution de secours
RFID	inexistante	nulle, ajout de <i>beacons</i>	$< 1m$	Musées
Infrarouge	inexistante	nulle, ajout de <i>beacons</i>	$< 1m$	Points d'intérêts
Ultrason	inexistante	nulle, ajout de <i>beacons</i>	$< 1m$	Points d'intérêts

TABLE 1.3 Tableau comparatif des signaux disponibles pour la localisation.

Récapitulatif

À titre comparatif, nous avons établi un tableau récapitulatif (tableau 1.3) de l'ensemble des signaux pouvant être utilisés. Pour la précision, comme nous l'avons indiqué, elle est particulièrement dépendante de l'infrastructure du bâtiment et de l'environnement. Nous proposons donc un ordre de grandeur des précisions pouvant être obtenues dans des bâtiments classiques (aéroport, supermarchés, etc.) sans traitement spécifique.

1.3 Limiter les besoins en infrastructure

Dans cette section, nous présentons des solutions pour limiter les besoins en infrastructure des solutions de localisation par signal RF. En effet, la plupart des solutions proposent de rajouter des points d'accès pour augmenter la précision, mais il s'agit d'une solution difficile à envisager pour l'ensemble des bâtiments dans le monde. Pour les applications nécessitant une précision d'une dizaine de mètres sans avoir à installer d'infrastructure supplémentaire, un modèle de propagation *indoor* peut être adapté.

Comme le Wi-Fi est le signal le plus utilisé, notamment grâce à sa densité dans les bâtiments, nous nous concentrerons sur cette technologie, même si tous les éléments peuvent être adaptés pour les autres signaux.

1.3.1 Modèle déterministe

La principale limite de la localisation par trilatération est la relation entre la puissance reçue et la distance au point d'accès. Cela provient du fait que l'on a utilisé dans un premier temps pour l'*indoor* des modèles de propagation conçus pour l'extérieur, notamment pour une propagation de type LOS (*Line-Of-Sight*), c'est-à-dire en ligne de vue directe.

Bâtiment	C	γ
Résidentiel	2, 8	$5(\text{maison}) - 10(\text{appartement})$
Bureau	30	14
Commerce	22	6
Usine	21	—

TABLE 1.4 Tableau des valeurs de C et γ recommandés par l'ITU pour 2,4 GHz.

C'est l'équation *Free Space Path Loss* présentée plus haut qui est utilisée et généralisée, souvent exprimée en dB :

$$P_{\text{émise}} - P_{\text{reçue}} = 20 \log_{10}(d) + C \quad (1.5)$$

où d est la distance entre l'émetteur et le récepteur, et C est une constante sensée combiner les effets des différentes atténuations que peut subir le système en extérieur.

Il existe également d'autres modèles, plus complets, de propagation en extérieur, souvent empiriques, dédiés particulièrement aux réseaux cellulaires. En effet, les opérateurs cherchent à maximiser la couverture tout en minimisant le nombre de stations de base. Ces modèles différencient la propagation suivant l'environnement : campagne, ville, etc. Le même type d'études a été mené pour la propagation en intérieur, en séparant suivant le type de bâtiment : aéroport, bureaux, supermarchés, etc. Ainsi, la première adaptation à l'*indoor* est la modification du coefficient C suivant la nature du bâtiment. Par ailleurs, un autre modèle a également été proposé, le *Log-distance path loss model* qui ajoute un coefficient de *pathloss* γ pour accentuer l'effet de la perte en puissance due à un plus grand nombre d'obstacles en environnement intérieur. On obtient :

$$P_{\text{émise}} - P_{\text{reçue}} = 10\gamma \log_{10}(d) + C. \quad (1.6)$$

Empiriquement, l'ITU a proposé plusieurs valeurs de C et de γ pour des environnements et des fréquences précises [52]. Le tableau 1.4 retrace un exemple pour le Wi-Fi à 2,4 GHz.

Certains modèles prennent en compte la connaissance du plan du bâtiment pour déterminer de façon probabiliste le nombre de murs que le signal a pu traverser, en utilisant un algorithme de détection de lignes sur un plan - appelé *Cohen-Sutherland*

line-clipping algorithm [18]. En effet, la traversée d'un mur atténue le signal d'une certaine puissance, suivant son matériau et sa profondeur, comme le montre la figure 1.9. Ce modèle, appelé *Wall Attenuation Factor* (WAF) ajoute deux paramètres :

$$P_{\text{émise}} - P_{\text{reçue}} = 10\gamma \log_{10}(d) + \begin{cases} w * WAF & \text{si } w < W_{max} \\ W_{max} * WAF & \text{si } w \geq W_{max} \end{cases} \quad (1.7)$$

où w est le nombre de murs rencontrés et WAF un facteur d'atténuation estimé en fonction de la nature du bâtiment. Les expérimentations [18] ont montré que, au-delà de W_{max} , les murs n'atténuent plus suffisamment le signal pour que cela soit pris en compte.

En plus de la connaissance du nombre de murs, le principal problème du WAF est l'estimation des valeurs W_{max} et WAF car les murs n'ont pas tous la même nature ni la même profondeur. L'atténuation du signal en dépendra.

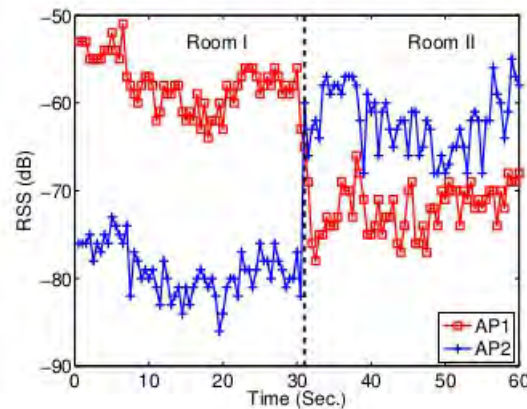


FIGURE 1.9 Courbe de perte de puissance du signal lors de la traversée d'un mur séparant deux pièces. Le point d'accès 1 est dans la pièce 1 et le 2 dans la pièce 2 [4].

Des modèles totalement empiriques plus complexes, comme le modèle fourni par Ericsson [53] à ses clients, existent, et expriment simplement la perte en puissance (en dB) comme une fonction polynomiale des logarithmes de la distance et de la taille des antennes. Ce modèle est plus adapté à la propagation de signaux cellulaires, où l'on peut maîtriser les paramètres de l'antenne.

1.3.2 Éviter la connaissance des points d'accès

Dans les solutions présentées dans cette section, il est nécessaire d'avoir des informations sur les points d'accès :

- leur position dans le bâtiment : deux coordonnées dans le cas à deux dimensions, trois coordonnées dans le cas à trois dimensions (la plupart des solutions existantes se limitent à deux dimensions).
- la puissance d'émission : bien qu'on puisse en déterminer un minimum et un maximum, et qu'il existe une puissance par défaut, celle-ci n'est pas précise.

L'ensemble de ces informations n'est pas facile à obtenir. Les points d'accès sont souvent mobiles et contrôlés par le client. Google a notamment cartographié les points de présence des points d'accès grâce à ses utilisateurs et à ses Google cars qui ont sillonné la France, pour déterminer la position des points d'accès à partir de la position GPS des mobiles. Cependant, des changements peuvent apparaître à tout moment. C'est la raison pour laquelle l'application de localisation par Wi-Fi présente sur les smartphones Android ne donne qu'une position approximative, souvent une première acquisition, avant de pouvoir obtenir le signal GPS. L'application continue d'utiliser les données de localisation des utilisateurs le désirant pour améliorer et mettre à jour sa connaissance de la topologie des points d'accès.

Les auteurs de [54] proposent de se passer de ces informations. Supposons un bâtiment où il y a m points d'accès et n utilisateurs. Tous connaissent la puissance reçue de chacun des points d'accès : on a une équation reliant la distance entre chaque mobile j et chaque point d'accès i , la puissance reçue par j de i , la puissance d'émission du point d'accès i et le coefficient γ , supposé constant pour chaque point d'accès - ce qui est inexact étant donné qu'il dépend de chaque trajet $i - j$.

$$P_{\text{émission}_i} - P_{\text{reçue}_{i,j}} = 10\gamma_i \log_{10}(d_{i,j}) + C. \quad (1.8)$$

Il y a donc mn équations. Or, il y a $4m + 2n$ inconnues en deux dimensions (les deux coordonnées de position pour les deux équipements, ainsi que γ et la puissance émise en ce qui concerne les points d'accès). Comme le nombre d'équations augmente de façon quadratique et le nombre d'inconnues linéairement, si m et n sont suffisamment grands, on peut en déduire toutes les positions, relativement à une seule position (acquise par GPS, par exemple). Les smartphones envoient leurs données (les puissances des signaux reçus) à un serveur qui centralise l'information, calcule toutes les positions, et renvoie aux équipements leur position.

1.3.3 Sans infrastructure : la solution du *crowdsourcing*

Plusieurs solutions de la littérature proposent l'utilisation du *crowdsourcing* [55–57] afin d'améliorer la précision de la localisation en faisant du partage d'information ou bien encore de se libérer de la contrainte de l'infrastructure. En effet, les smartphones sont capables d'émettre en Wi-Fi entre eux (Wi-Fi direct par exemple) ou d'agir en points d'accès. Les contraintes sont importantes car cela oblige l'utilisateur à couper son accès Internet pendant cette phase pour aider les autres utilisateurs à se localiser. De plus, il reste le problème inhérent au *crowdsourcing* : plus le nombre d'utilisateurs est grand, meilleure est la précision, mais pour attirer des utilisateurs, il faut leur proposer une qualité importante dès le début. Les auteurs de Social-Loc [57] estiment le gain de précision entre 22% et 37%.

1.4 Empreintes du bâtiment : le pré-déploiement

Afin d'améliorer la précision de la localisation dans des bâtiments particuliers, des travaux ont été menés pour proposer du pré-déploiement. Cette technique est adaptée aux bâtiments dans lesquels des opérations de manutention sont possibles, mais qui restent stables dans le temps, c'est-à-dire que les murs ou obstacles sont très peu déplacés. Par exemple, les bâtiments comme les aéroports ou les gares peuvent utiliser ce type de méthodes. Plutôt que de proposer à l'utilisateur de se positionner, c'est le système qui cherche la position de l'utilisateur.

1.4.1 Principe du *fingerprinting*

Le pré-déploiement d'un système de localisation dans un bâtiment consiste en une prise d'empreintes, ou *fingerprinting*. Avant le lancement de la solution de localisation, une équipe de techniciens entre dans le bâtiment où l'on souhaite pouvoir localiser les smartphones. Ils s'arrêtent en certains points aux positions connues - plus ou moins régulièrement espacés -, et scannent les signaux environnants, cf figure 1.10. Ce scan est sauvegardé et envoyé à un serveur.

On obtient pour chaque scan $i \in [1..I]$ localisé dans l'espace un scan R_i :

$$R_i = [r_{i,1}, \dots, r_{i,J}] \quad (1.9)$$

où J est le nombre de points d'accès présents dans le bâtiment, et $r_{i,j}$ la puissance de signal reçue.

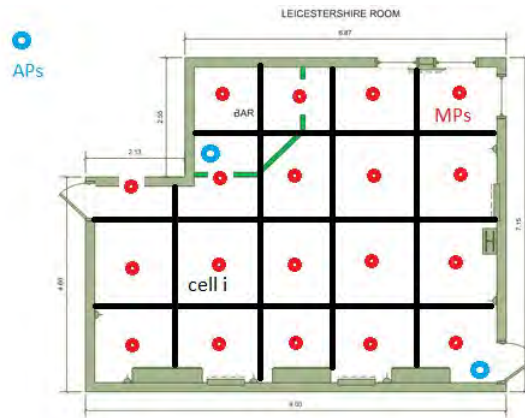


FIGURE 1.10 Exemple de prise d’empreinte d’une pièce. Les points rouges correspondent à l’endroit de chaque scan. En bleu, les points d’accès captés.

Cette matrice R est stockée sur un serveur. Notons que l’on a parfois plusieurs mesures par point en faisant varier les positions relatives du smartphone et de l’utilisateur pour une même localisation [58]. En effet, l’orientation du smartphone par rapport au point d’accès change [56] la valeur de la puissance de signal reçue. C’est la fin de la phase hors-ligne. Par la suite, quand un smartphone souhaite être localisé dans le bâtiment, il va réaliser son propre scan $S = [s_1, \dots, s_J]$ et l’envoyer vers le serveur, qui va comparer ce scan à l’ensemble des scans réalisés dans les positions connues de la matrice R . La position estimée du smartphone sera celle dont le scan R_i réalisé *offline* est le plus semblable à S .

Cette solution d’empreintes permet de s’affranchir totalement de la modélisation du canal, qui est très difficile en intérieur. En effet, aucune supposition sur la distance par rapport aux points d’accès n’est faite : il s’agit seulement de la comparaison de signaux. Les principaux problèmes de ces techniques résident dans la phase de pré-déploiement. D’une part, elle est très coûteuse et ne peut pas être généralisée à l’ensemble des bâtiments. D’autre part, tout changement dans le bâtiment nécessite la mise en œuvre d’une nouvelle phase d’empreintes, étant donné que la propagation du signal aura changé. Enfin, même de petits changements dans la densité de personnes présentes dans la pièce au moment du scan ou de température peut également dégrader les performances. La précision obtenue est fortement dépendante de la phase de collecte : nombre de points, qualité de la mesure, etc. Il est communément admis qu’une précision entre 5 et 10 m peut être obtenue.

1.4.2 Estimation déterministe de la position

Pour déterminer la position la plus probable du smartphone après le scan parmi les positions de la matrice, les premières solutions utilisent une méthode déterministe,

tels que les K plus proches voisins (*K-Nearest Neighbor* ou KNN) [59]. Les auteurs de WILL [4] ou de [20, 21] proposent une méthode euclidienne. Ils calculent un coefficient de similarité C_i pour chaque position stockée, qui est la distance euclidienne entre le scan en position i et le scan *online* :

$$C_i = \sqrt{\sum_{j=1}^J (s_j - r_{i,j})^2}. \quad (1.10)$$

La position l estimée est alors celle qui maximise le coefficient de similarité :

$$l = \operatorname{argmax}_{i \in [1..I]} (C_i). \quad (1.11)$$

Cette estimation est particulièrement sensible aux changements dans le bâtiment dans la mesure où elle considère comme indépendants l'ensemble des scans effectués. Or, deux scans réalisés dans des positions proches seront semblables, ce qui risque de causer des erreurs de positionnement.

1.4.3 Estimation probabiliste de la position

Pour éviter la sensibilité aux changements de propagation de signal dans le bâtiment, des méthodes probabilistes ont été proposées [60, 61], la plus efficace et connue étant Horus [62]. Plutôt que de réaliser un scan ponctuel, à chaque position i plusieurs scans successifs sont effectués. On obtient une observation O_i formée de l'ensemble des θ observations : $O_i = [R_i(1), \dots, R_i(\theta)]$. Cette observation est considérée comme représentative d'une distribution des RSS au point i . Dès lors, on peut estimer la probabilité de recevoir un scan S lorsqu'on est au point l :

$$Pr(S|l) = \prod_{j=1}^J Pr(s_j|l, j) \quad (1.12)$$

où $Pr(s_j|l, j)$ est la probabilité d'obtenir un RSS s_j provenant de j quand on est au point l , et calculé à partir de la matrice O . On peut alors estimer la probabilité *a posteriori* d'être en position l conditionné par un scan S :

$$Pr(l|S) = \frac{Pr(S|l)Pr(l)}{\sum_{i=1}^I Pr(S|i)Pr(i)} \quad (1.13)$$

où l'on suppose connue en chaque point la probabilité qu'un smartphone soit dans la position i . On peut alors retrouver la position l :

$$l = \operatorname{argmax}_{i \in [1..I]} (Pr(i|S)). \quad (1.14)$$

Cette méthode est plus robuste, mais oblige à une phase de pré-déploiement d'autant plus longue et coûteuse. WILL [4] propose de s'affranchir du pré-déploiement en combinant deux solutions de localisation. La première est une méthode de type déterministe, qui permet d'obtenir une localisation imprécise. Elle représente aussi un scan (un point de l'empreinte du bâtiment) pour la seconde, fondé sur du *fingerprinting*, grâce à l'ensemble des scans précédemment obtenus par l'ensemble des utilisateurs.

1.4.4 Améliorations liées à la navigation

Un grand nombre de travaux ont ensuite suivi ces recherches pour améliorer la précision de localisation de ces méthodes. Sans en faire une liste complète, une évolution majeure réside dans l'opportunité d'utiliser le trajet de l'utilisateur. Les positions successives de l'utilisateur ne sont pas indépendantes. Ainsi, il est possible de corriger une position aberrante de l'utilisateur en prenant en compte de ces positions précédentes, par exemple à l'aide d'un filtre de Kalman [63, 64].

1.5 Modifications des smartphones

Certaines applications liées à la localisation *indoor* requièrent une précision accrue par rapport à ce qu'on a pu voir jusqu'ici, toutes de l'ordre de quelques mètres. La solution envisagée dans la littérature pour résoudre ce problème réside dans la modification des smartphones pour obtenir des capteurs de meilleure qualité.

1.5.1 Pour le *Time of arrival*

Plusieurs entreprises, notamment la start-up française BeSpoon [65] ont montré la possibilité d'intégrer au smartphone un *chip* UWB (*Ultra-Wide Band*). Cette puce permet de réaliser une transmission d'impulsions de très courte durée, de l'ordre de la nanoseconde. Cette technologie, amenée à remplacer le Bluetooth dans la mesure où elle permet d'atteindre de haut débit sur des distances courtes, peut être utilisée pour résoudre le problème de l'imprécision de la technique TOA pour les autres signaux. Comme les émissions sont très courtes, elle est moins sensible aux multi-trajets notamment. Ainsi, les systèmes commerciaux prétendent pouvoir atteindre une résolution exceptionnelle, de

l'ordre de la dizaine de centimètres grâce à cette technologie. De la même façon que le Bluetooth, elle nécessite l'installation de *beacons* dans les pièces où l'on souhaite pouvoir proposer la localisation. Or, la portée de l'UWB étant très faible, un grand nombre de *beacons* est nécessaire.

1.5.2 Pour connaître l'angle d'arrivée du signal

Plutôt que d'essayer de trouver la distance entre le point d'accès et les terminaux mobiles, certains travaux ont été menés pour se rapprocher de la méthode concurrente de la trilatération : la triangulation. Plus exactement, une première idée est simplement de ne pas déterminer seulement la distance entre le point d'accès et le smartphone, mais de trouver également la direction du signal. Ainsi, un seul point d'accès est nécessaire à la localisation, comme le montre la figure 1.11. Malheureusement, cette information ne peut pas être déterminée sans les informations de la couche physique. En revanche, les cartes Wi-Fi possèdent cette information. En effet, elles estiment continuellement la qualité du canal de transmission, appelée CSI (*Channel State Information*). Si l'émetteur envoie le symbole X et que le récepteur reçoit le symbole Y :

$$Y = H * X + n \quad (1.15)$$

où n représente le bruit. Le CSI est un estimé de la matrice H .

La connaissance de ce CSI permet de retrouver l'angle d'arrivée du chemin direct (ADP - *Angle of Direct Path*), même en cas de présence de multi-trajets. Les auteurs de [66] ont modifié la carte Wi-Fi d'un smartphone pour récupérer le CSI et implanté une solution de ce type, qui combine la distance et l'angle d'arrivée : ils atteignent des erreurs de quelques mètres.

D'autres travaux de recherche poursuivent la stratégie d'utilisation du CSI. C'est le cas des auteurs de [67] qui récupèrent les données du CSI pour réaliser un radar à synthèse d'ouverture (SAR - *Synthetic Aperture Radar*) grâce à la connaissance de l'angle d'arrivée. Cela permet une précision de l'ordre de la dizaine de centimètres. Pourtant, l'algorithme présenté demande aux smartphones d'être équipés de plusieurs antennes espacées d'une dizaine de centimètres, c'est la raison pour laquelle ils ont déployé leur système uniquement sur des tablettes.

Enfin Sextant [68] propose à l'utilisateur de pointer son téléphone vers le point d'accès Wi-Fi afin de calculer l'angle d'arrivée à l'aide du gyroscope interne au smartphone.

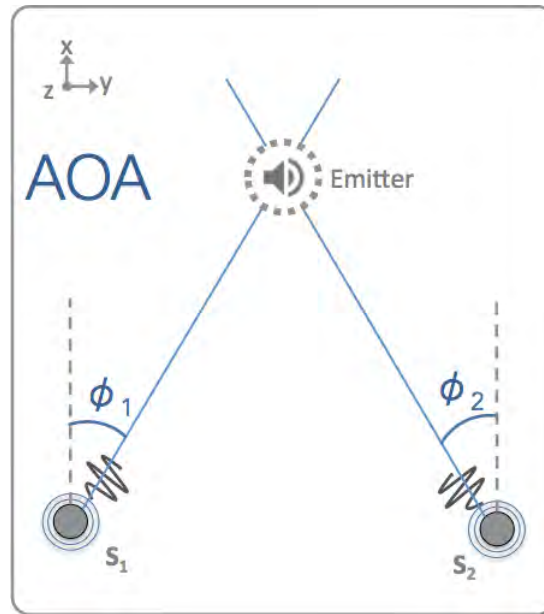


FIGURE 1.11 La station S_1 se localise en mesurant la distance à l'émetteur - par exemple à l'aide de la puissance reçue -, et en mesurant l'angle Φ_1 par rapport à l'antenne émettrice [5].

1.6 Positionnement, navigation ou détection d'environnement

Les systèmes de localisation présentés cherchent à positionner un smartphone dans le bâtiment ou proposent à l'utilisateur de se positionner grâce à son smartphone. Pourtant, il ne s'agit pas de l'application principale de la localisation sur mobile : généralement, l'utilisateur cherche à suivre un trajet plutôt que seulement connaître sa position. Certaines techniques cherchent donc à proposer à l'utilisateur de smartphone de suivre son trajet, plutôt qu'une localisation ponctuelle. Pour les techniques déjà présentées, il a été proposé, par exemple, d'ajouter un filtre de Kalman, qui permet de lisser les positionnements successifs de l'utilisateur, grâce à de la prédiction et de la mise-à-jour [63, 64]. Inversement, si le système souhaite une localisation ponctuelle de l'utilisateur, il n'est pas toujours essentiel de déterminer sa position précise, mais il peut être suffisant de connaître la pièce dans laquelle il se trouve.

Ces nouvelles applications tirent profit de l'intelligence de nos smartphones : de plus en plus, les mobiles de nouvelle génération sont équipés de capteurs de toute sorte. Tous les smartphones comprennent magnétomètres, capteur de luminosité, centrale inertielle

(accéléromètres et gyroscopes), microphone, caméra. S'ils ne sont pas toujours d'une précision absolue, ils sont des outils pour étendre les applications possibles. Cette section présente les solutions qui reposent sur ces nouveaux capteurs et qui répondent précisément aux attentes de la localisation *indoor*.

1.6.1 Localisation par rapport à un environnement

Les auteurs de [69] proposent de corriger les erreurs de localisation des méthodes précédentes en différenciant deux pièces proches. L'exemple type est celui d'un centre commercial, où deux boutiques attenantes peuvent être de nature différente : une librairie et un pub par exemple. Ces deux endroits ont des ambiances très différentes : niveau sonore, luminosité, mouvements des utilisateurs, etc. Connaître l'ambiance, c'est donc savoir si on est d'un côté ou de l'autre du mur - ce qui donne une marge d'erreur possible pour les solutions précédentes, dont les précisions sont de l'ordre du mètre.

Pour cela, les auteurs ont constitué une base de données de l'ambiance dans différents types d'environnement : magasins, boutiques, librairies, pubs, etc. Cette base de données contient notamment les accélérations mesurées (mesurant la quantité de mouvement des utilisateurs), le son ambiant, la couleur dominante et la luminosité. Ainsi, lorsque l'on souhaite localiser un utilisateur, il suffit de comparer l'ambiance captée par son téléphone avec celle de la base de données, de la même façon que pour le *fingerprinting* Wi-Fi. Dans 87% des cas, SurroundSense détecte correctement le magasin dans laquelle se trouve l'utilisateur.

1.6.2 Principes de *deadreckoning*

La navigation à l'estime (ou *deadreckoning*) a été utilisée depuis toujours pour déterminer sa position. Christophe Colomb utilisait ce principe pour connaître sa direction. Le principe en est simple : connaissant sa position initiale, l'utilisateur peut déduire sa position finale grâce à ses mouvements successifs. À l'inverse des autres systèmes de localisation présentés, on ne peut pas estimer la précision d'un système à tout moment, étant donné que l'erreur croît en fonction du déplacement. En effet, chaque erreur est conservée et cumulée à chaque estimation.

Ce système a été proposé depuis l'apparition des accéléromètres dans les smartphones. Connaissant la position initiale d'un utilisateur à l'entrée d'un bâtiment (acquis par GPS par exemple), les accéléromètres peuvent nous permettre de suivre le mouvement du smartphone et de calculer sa position à chaque instant par une simple double intégration - en supposant la vitesse initiale également connue. La figure 1.12 montre que cette solution

simple ne fonctionne pas, à cause de l'imprécision des accéléromètres des smartphones, et de la dérive d'intégration (*integration drift*) : une petite erreur dans la donnée d'accélération va se répercuter fortement lors de l'intégration et sera conservée lors de toutes les positions suivantes, puisqu'elles sont calculées à partir des précédentes.

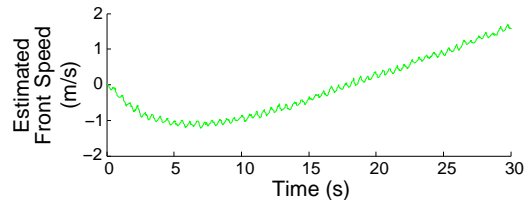


FIGURE 1.12 La dérive d'intégration ne permet pas d'utiliser les données pures des accéléromètres pour suivre les mouvements du smartphone. Ici, l'utilisateur se déplace à une vitesse constante de $0,5\text{m/s}$, mais la vitesse estimée par intégration ne l'est pas du tout.

1.6.3 Comptage de pas

En attendant une amélioration significative de la précision des accéléromètres des smartphones, la recherche d'une solution de localisation à partir des valeurs d'accélération s'est rapidement essouffée. Pourtant, notamment relancées par Google et Apple, des applications utilisant les accéléromètres se sont développées, comme le comptage de pas. Ces applications utilisent la forme typique du signal d'accélération pour en déduire que l'utilisateur a fait un pas. La figure 1.13 montre la répétition du signal d'accélération. Chaque pic correspond à un pas.

La présence d'un pic ne suffit pas pour être convaincu de la présence d'un pas. Il se peut que le pic ait été provoqué par un mouvement de l'utilisateur du smartphone par exemple. Décryptons davantage le signal d'accélération au cours d'un pas, comme sur la figure 1.14. Lorsque le pied quitte le sol (jambe en arrière) à t_0 , l'accélération augmente régulièrement jusqu'à rejoindre la verticale du corps à t_1 , puis diminue jusqu'à ce que le pied heurte le sol (jambe en avant). À ce moment, l'accélération subit une secousse marquée par une double forte accélération en t_2 et t_3 . Le pied se pose ensuite entre t_4 et t_5 .

Les principales propositions de comptage de pas de la littérature [55, 70, 71], plus efficaces que les solutions incluses dans les smartphones Android ou Apple, reconnaissent ainsi les formes de pas grâce à un traitement du signal plutôt simple en détectant le moment où le pied heurte le sol :

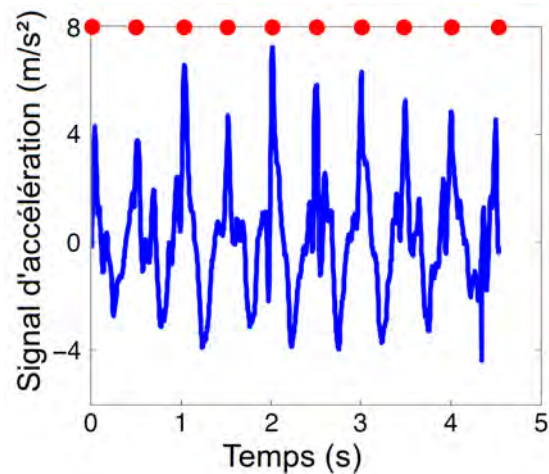


FIGURE 1.13 Mesure de l'accélération en fonction du temps : chaque pic point rouge correspond à un pas.

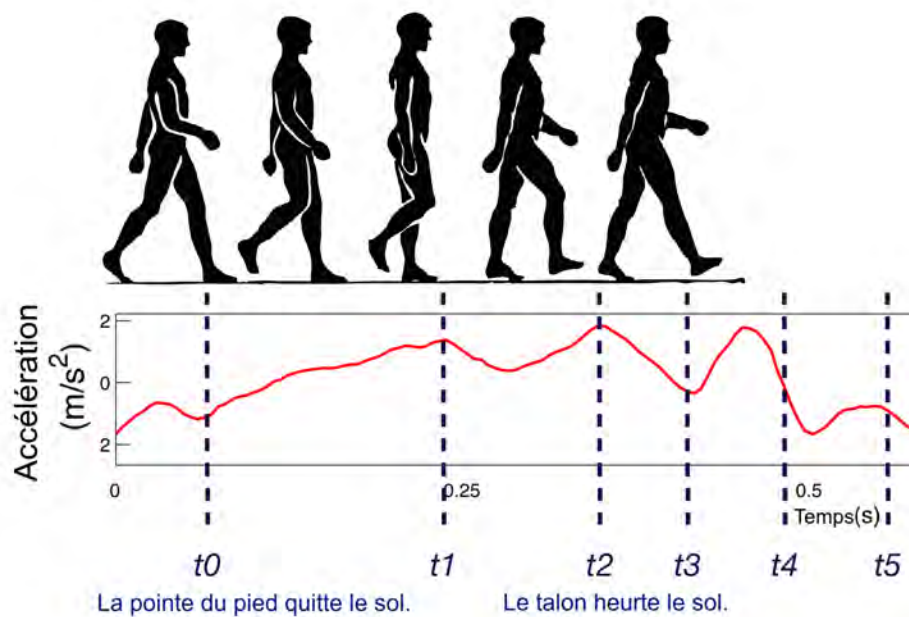


FIGURE 1.14 La forme de l'accélération pour un pas est commune à tous les êtres humains : elle est surtout marquée par un double pic au moment où le talon heurte le sol.

1. le signal d'accéléromètre est passé au travers d'un filtre passe-bas afin d'enlever les mouvements aléatoires parasites à haute fréquence du smartphone ;
2. deux minima locaux sont recherchés ;

3. entre ces minima, un maximum local est recherché ;
4. enfin, dans ces travaux, les auteurs vérifient certains critères qui doivent être satisfaits pour valider un pas. Suivant les solutions, plusieurs critères sont proposés : vérifier que la différence entre le maximum et les minima est plus grande qu'un seuil ; la durée entre deux maxima (donc deux pas) est comprise entre deux valeurs (typiquement entre 0,2 s et 1s) ; etc.

La détection d'un "faux" pas est toujours possible, mais sa probabilité est réduite. Dès lors, cette technique a été envisagée pour la mesure de la distance parcourue par un utilisateur de smartphone. Pour cela, il suffit de connaître la taille d'un pas. De nombreuses études ont été menées pour la déterminer, suivant un grand nombre de paramètres. Il est apparu que la taille d'un pas moyen chez un individu varie suivant de très nombreux paramètres : la taille, le poids, l'âge, le genre, etc. [70, 72]. Des modèles empiriques à partir de ces paramètres [73] ont été proposés mais obligent l'individu à fournir un grand nombre d'informations sur lui-même (taille, poids, longueur de jambe, etc.). Par ailleurs, certaines solutions ont proposé de mesurer la taille d'un pas moyen d'un individu par une demande spécifique à l'utilisateur - on lui demande de marcher sur une distance précise, et on compte le nombre de pas qu'il a effectué. Pour une application destinée au grand public, cela semble compliqué. Pour contourner le problème, il est possible de mesurer de façon opportune la taille de pas, par exemple lorsque l'utilisateur est en extérieur et que l'on capte un signal GPS. On a vu pourtant que le signal GPS n'était pas très précis (quelques mètres), ce qui risque de limiter la qualité de la mesure.

Quelle que soit la solution permettant d'estimer la taille moyenne de pas de l'utilisateur, le résultat ne sera pas parfait. En effet, la taille d'un pas chez un individu varie également fortement [71].

Une fois la distance parcourue estimée, il faut également connaître la direction du pas. Là encore, le gyroscope n'est pas assez précis pour intégrer simplement la vitesse de rotation, comme le montre la figure 1.15, notamment à cause du mouvement de balancier du corps pendant le pas. Les auteurs de [74] ont montré qu'il était plus avantageux de ne regarder la direction de marche qu'au moment où le talon se pose à terre ; ils obtiennent une précision absolue (dans le référentiel terrestre) inférieure à huit degrés, quelle que soit la position du smartphone.

1.6.4 En utilisant la caméra

Plusieurs solutions proposent également d'utiliser la caméra, sans doute le capteur le plus performant des smartphones, afin de suivre l'utilisateur dans ces mouvements.

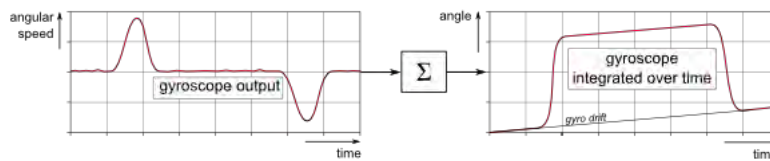


FIGURE 1.15 Les gyroscopes souffrent aussi de la dérive d'intégration lorsqu'on veut retrouver la direction de marche [6]. Lorsque l'utilisateur tourne de 90° puis retrouve sa position, la direction estimée par le gyroscope est biaisée.

Luxapose [75] utilise des diodes LED positionnées au sol ou au plafond d'un bâtiment. La caméra repère les LED et suit le trajet de l'utilisateur à travers la succession de LED. La précision obtenue est centimétrique, mais elle nécessite évidemment une infrastructure particulièrement importante. Cette solution est adaptée, par exemple, pour les issues de secours ou des trajets bien déterminés sur lesquels des LED peuvent être placées le long du parcours.

Travi-Navi [76] proposent un système de navigation pour mal-voyants. Hors-ligne, une phase d'apprentissage par empreintes est réalisée ; cette fois, c'est la vidéo de chaque trajet possible dans le bâtiment qui est récupérée. Lorsqu'un utilisateur arrive dans le bâtiment, les images captées sont comparées aux empreintes afin de retrouver la position et le chemin pris par l'utilisateur. Cette solution, en plus de demander des efforts de pré-déploiement importants, consomme une grande quantité d'énergie, dans la mesure où la caméra allumée en continu fait décroître le niveau de batterie rapidement.

1.7 Notre positionnement

Dans cette section, nous positionnons les objectifs de la thèse parmi l'ensemble des solutions proposées dans la littérature. Nous proposons par ailleurs une solution hybride, qui combine plusieurs solutions et constitue une base pour obtenir une application de localisation globale et fonctionnelle quels que soient les besoins.

1.7.1 Limiter les infrastructures et les modifications d'équipements

La solution de localisation que nous souhaitons obtenir doit être disponible le plus facilement possible. Par conséquent, nous souhaitons utiliser les caractéristiques propres d'un smartphone du marché : sans modification, sans adjonction d'équipement et sans amélioration des capteurs. Par ailleurs, nous souhaitons être complètement libres des contraintes d'infrastructure et de pré-déploiement. Pour cela, nous optons pour une solution de type *deadreckoning*, de sorte qu'aucune communication ou aucun point d'accès

ne soit nécessaire pour effectuer la localisation. Cette méthode peut être complétée par un modèle déterministe de localisation par les signaux RF, par exemple Wi-Fi. Notre solution doit permettre la navigation en intérieur, plus que le positionnement. Enfin, nous voulons limiter l'intervention de l'utilisateur, et notamment ne demander aucune information sur lui (taille de pas, taille, poids, etc.).

La figure 1.16 présente une classification des solutions existantes et met en évidence la zone vide que nous souhaitons remplir pour répondre aux exigences présentées.

Légende

- **Système** : nom générique du système ou de l'article de recherche ;
- **Couverture** : type de couverture - *outdoor*, locale (*indoor*), mondiale (*indoor*) ;
- **Signal** : le type de signal nécessaire ;
- **Infrastructure** : infrastructure nécessaire - dédiée (installation de *beacons* spécifiques), connue (connaissance de l'infrastructure, par exemple la position des points d'accès, nécessaire), existante (se fonde sur l'infrastructure existante), aucune (aucun besoin spécifique), crowdsourcing (communication entre mobiles) ;
- **Pré-déploiement** : le type de pré-déploiement nécessaire : intensif (installation et empreinte), limité (installation seule), aucun ;
- **Mobile** : l'équipement utilisé - modifié, *off-the-shelf* (du marché) ;
- **Type de localisation** : naviguer (trajectoire), se positionner (par l'utilisateur), positionner (par un serveur distant), environnement ;
- **Précision** : un ordre de grandeur de la précision ;
- **Application** : un exemple typique d'application.

1.7.2 Architecture d'une solution hybride

Le positionnement que nous avons choisi n'est en aucun cas incompatible avec les solutions existantes de la littérature. Ainsi, une application stable, robuste et surtout capable de répondre à l'ensemble des applications de la localisation *indoor* pourrait combiner plusieurs solutions indépendantes. La figure 1.17 propose une architecture qui suit cet objectif. Notre solution de navigation permet d'obtenir la trajectoire de l'utilisateur (en vert) et est présentée explicitement dans le chapitre II. En parallèle, une solution de localisation par modèle déterministe [18, 53] peut être utilisée pour permettre au smartphone de se positionner. Pour des bâtiments spécifiques dans lesquels une localisation précise est nécessaire, une phase d'empreintes [62] peut être utilisée au préalable. Ces solutions permettant de déterminer la position de l'utilisateur, elles ne dépendent pas du temps de trajet. Elles peuvent donc permettre de corriger les éventuelles

erreurs de notre solution de navigation [4] pour obtenir une localisation plus précise et plus durable. Par ailleurs, connaissant la position de l'utilisateur, il est nécessaire de l'identifier vis-à-vis de son environnement : le bâtiment. Il peut être fourni par le propriétaire, mais ce n'est pas toujours le cas. Le chapitre III propose une solution pour construire le plan du bâtiment au cours du trajet de l'utilisateur grâce aux capteurs d'ambiance de son smartphone. L'application peut alors tracer le trajet de l'utilisateur avec le plan du bâtiment en fond. Il peut également lui proposer un trajet jusqu'à sa destination.

Système	Couverture			Signal	Infrastructure	Pré-développement		Mobile		Type de localisation		Précision	Application
	outdoor	indoor	locale			mondiale	aucun	limité	off-the-shelf	off-the-shelf	se positionner		
GPS		outdoor		GPS	déclée	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	8m	en extérieur	
GPS assisté [40]		outdoor		GPS	déclée	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	1m	en extérieur précis	
Wi-Fi ToA [19, 46]		locale		Wi-Fi	déclée	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	10m	localisation de partout	
RADAR (modèle Wi-Fi) [18]		mondiale		Wi-Fi	connue	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	8m	localisation de partout	
GSM [22]		mondiale		GSM	connue	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	10m	localisation de secours	
Bluetooth Low Energy [22, 23]		locale		Bluetooth	déclée	limité	limité	off-the-shelf	off-the-shelf	se positionner	5m	événements courte durée (congrès)	
Radio FM [49]		mondiale		FM	connue	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	15m	localisation de secours	
RFID [17]		locale		RFID	déclée	limité	limité	off-the-shelf	off-the-shelf	se positionner	<1m	bâtiment administré (musées)	
Infrarouge [50]		locale		Infrarouge	déclée	limité	limité	off-the-shelf	off-the-shelf	se positionner	<1m	bâtiment administré (musées)	
Ultrasound [51]		locale		Ultrasound	déclée	limité	limité	off-the-shelf	off-the-shelf	se positionner	<1m	bâtiment administré (musées)	
Social-Loc (Wi-Fi crowdsourcing) [57]		mondiale		Wi-Fi	crowdsourcing	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	6m	localisation de partout	
WILL (fingerprinting Wi-Fi déterministe et complet de pas) [4]		mondiale		Wi-Fi	existante	intensif	intensif	off-the-shelf	off-the-shelf	positionner	5m	localisation de partout	
Horus (fingerprinting Wi-Fi probabiliste) [62]		mondiale		Wi-Fi	existante	intensif	intensif	off-the-shelf	off-the-shelf	positionner	5m	localisation de partout	
EZ (fingerprinting Wi-Fi sans connaissance de l'infrastructure) [54]		mondiale		Wi-Fi	existante	intensif	intensif	off-the-shelf	off-the-shelf	positionner	7m	localisation de partout	
BeSpoon (UWB) [65]		locale		UWB	déclée	limité	limité	modifié	modifié	se positionner	<1m	localisation précise	
Angle of Arrival avec CSI [66]		mondiale		Wi-Fi	connue	aucun	aucun	modifié	modifié	se positionner	<1m	localisation précise	
SAR TOA [67]		mondiale		Wi-Fi	connue	aucun	aucun	modifié	modifié	se positionner	<1m	localisation précise	
Sixtant (Angle of Arrival avec pointage de l'utilisateur) [68]		mondiale		Wi-Fi	connue	aucun	aucun	off-the-shelf	off-the-shelf	se positionner	5m	utilisateur géré (employé)	
Tracking Wi-Fi [63]		mondiale		Wi-Fi	connue	aucun	aucun	off-the-shelf	off-the-shelf	naviguer	5m	navigation indoor	
SurroundSense [69]		locale		aucun	aucune	aucun	aucun	off-the-shelf	off-the-shelf	environnement	87% des pièces	correction de la localisation	
Accéléromètres et Gyroscopes : intégrations		mondiale		aucun	aucune	aucun	aucun	off-the-shelf	off-the-shelf	naviguer	erreur exponentielle	navigation indoor	
Comptage de pas		mondiale		aucun	aucune	aucun	aucun	off-the-shelf	off-the-shelf	naviguer	>1m tous les 100m	navigation indoor	
Zee (comptage de pas et crowdsourcing Wi-Fi) [55]		mondiale		Wi-Fi	crowdsourcing	aucun	aucun	off-the-shelf	off-the-shelf	naviguer	5m	navigation indoor	
Luxapose (LED) [75]		locale		aucun	déclée	limité	limité	off-the-shelf	off-the-shelf	naviguer	<1m	malvoyants	
Travi-Navi (empreintes vidéo et accéléromètres) [76]		locale		aucun	déclée	intensif	intensif	off-the-shelf	off-the-shelf	naviguer	2m	malvoyants	
Notre positionnement		mondiale		aucun	aucune	aucun	aucun	off-the-shelf	off-the-shelf	navigation	<5m sur tous les trajets indoor, <3m en moyenne	navigation indoor	

FIGURE 1.16 Classification des solutions de localisation *indoor*.

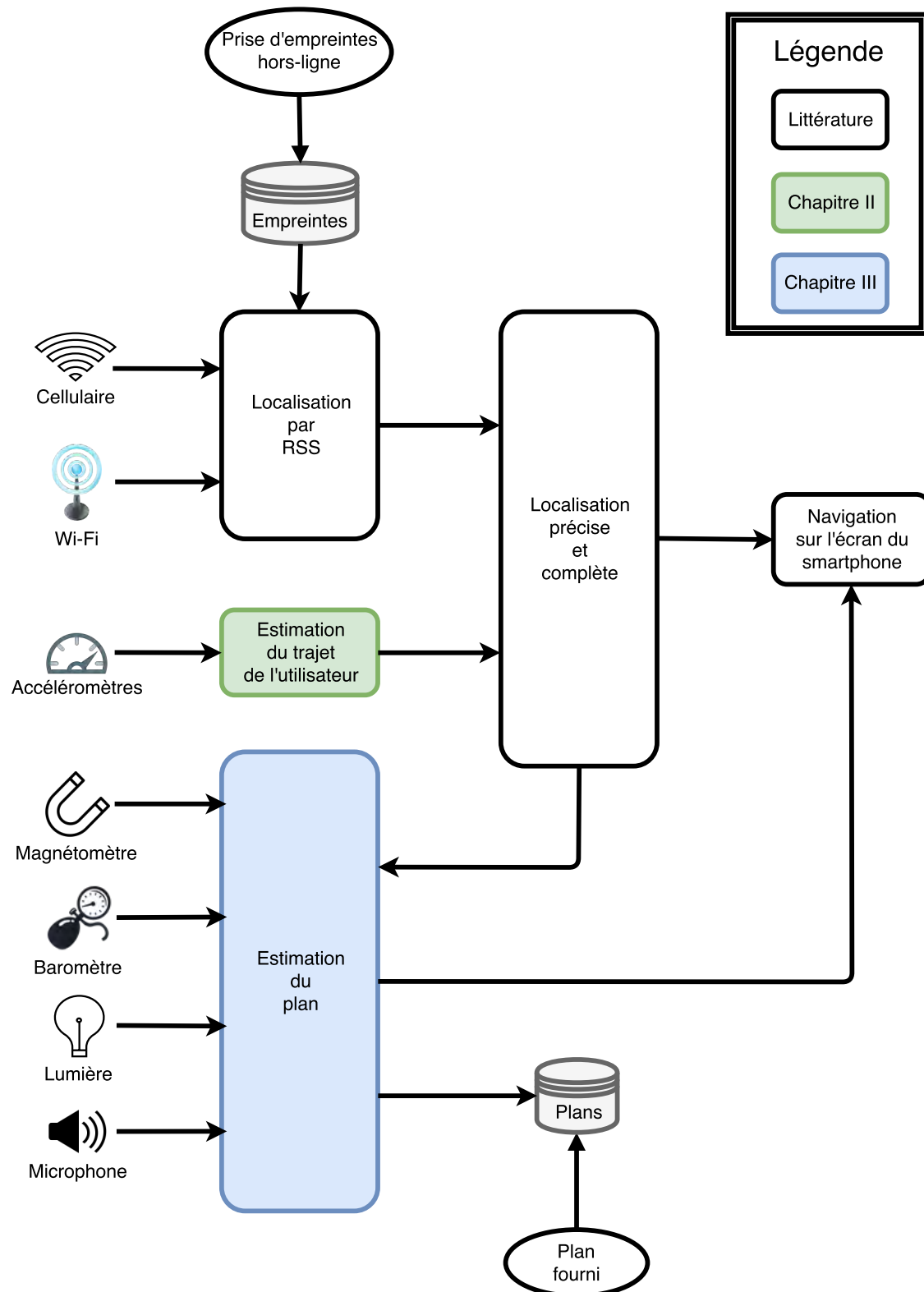


FIGURE 1.17 Notre proposition d'architecture hybride.

ACRUX : UNE SOLUTION DE NAVIGATION PAR DIFFÉRENCIATION DE LOCOMOTIONS

Dans ce chapitre, nous présentons Acrux, notre solution de navigation sur smartphone. Il s'agit du premier système qui permette de conserver une précision acceptable pour la navigation dans la plupart des bâtiments, tout en n'utilisant qu'un *fix* (une position initiale connue, à l'entrée du bâtiment) et les capteurs de mouvement des smartphones du marché. En reprenant le principe de navigation à l'estime, Acrux est donc une application autonome et indépendante de l'infrastructure, sans besoin de pré-déploiement.

Jusqu'ici, la navigation à l'estime a été mise de côté pour les smartphones à cause de l'imprécision des gyroscopes intégrés et de la dérive d'intégration. Il est *a priori* impossible d'éliminer complètement cet effet. C'est pourquoi certaines solutions proposent de ne pas prendre en compte la valeur remontée par les capteurs, mais d'observer la forme du signal pour compter les pas de l'utilisateur. Cela nécessite une nouvelle entrée : la taille d'un pas. Celui-ci variant énormément dans le temps et en fonction de l'utilisateur, les solutions de la littérature proposent des systèmes de recalibration périodique. Que ce soit grâce à des données entrées par l'utilisateur ou en utilisant d'autres systèmes de localisation, ces recalibrations sont un frein important pour le passage à l'échelle.

Acrux permet de contourner ce problème en analysant les différents types de déplacement d'un utilisateur humain, appelés **locomotions** [77, 78].

Dans ce chapitre, nous présenterons l'architecture de la solution, avant de détailler les trois blocs principaux : la détection des locomotions, la mesure de cadence des pas et le modèle pour le passage à trois dimensions. Nous analyserons ensuite les performances du système et nous donnerons ses limites.

2.1 Présentation d'Acrux

2.1.1 Cadre applicatif

Alice navigue à l'aide du GPS de son smartphone jusqu'à ce qu'elle perde le signal : la couverture satellite est absente dans sa zone - un bâtiment, une station de métro, etc. L'application Acrux récupère la dernière position obtenue puis continue de la suivre tout au long de son trajet jusqu'à sa destination, même en intérieur. L'application est autonome : elle n'a aucun besoin d'infrastructure, de pré-déploiement et ne demande pas de recalibration. Elle repose uniquement sur les capteurs internes du smartphone d'Alice.

Acrux a donc pour objectif de proposer une solution de navigation *indoor* qui puisse être utilisée dans n'importe quel bâtiment, même inconnu. L'application suit la trajectoire de l'utilisateur du smartphone qui se déplace à pied. Elle propose à l'utilisateur de pouvoir connaître sa position en n'importe quel endroit du bâtiment en fonction d'une position initiale connue, par exemple l'entrée du bâtiment.

2.1.2 Les capteurs de mouvement pour la navigation

Avec l'augmentation du nombre de capteurs embarqués sur les smartphones, l'idée d'utiliser les capteurs paraît naturelle. En effet, les smartphones incluent désormais trois types de capteurs, accessibles le plus souvent via des API (*Application Programming Interface*) fournies par le système d'exploitation (Android, iOS ou Windows) [79].

1. **Capteurs de mouvement** : il s'agit des capteurs qui analysent les mouvements du smartphone. Ensemble, ils forment une centrale inertielle de deux capteurs : gyroscope et accéléromètre, tous deux dans les trois axes du smartphone, comme le montre la figure 2.2a. Le gyroscope mesure la vitesse de rotation du smartphone (en rad/s) et l'accéléromètre son accélération (en m/s^2). On peut aussi inclure dans cette catégorie d'autres capteurs, comme le capteur de gravité, d'attitude ou d'accélération linéaire, mais ils ne sont pas des capteurs à part entière dans la mesure où leurs valeurs sont déterminées à partir de la centrale inertielle.
2. **Capteurs d'ambiance** : il s'agit des capteurs qui permettent de décrire l'environnement du smartphone. Il existe notamment : le capteur de champ magnétique à trois axes (en μT), le capteur de luminosité (en lux), le baromètre (qui mesure la pression atmosphérique, en hPa), le capteur de proximité (qui mesure la distance de l'objet le plus proche, en cm), le capteur d'humidité (qui mesure l'humidité de l'air, en g/m^3), le capteur de température (en Celsius). On peut ajouter à cette liste le microphone qui permet de calculer le niveau sonore ambiant, en dB.

3. **Émetteurs et récepteurs de signaux RF** : il s'agit des capteurs permettant aux smartphones de communiquer. Les smartphones incluent un récepteur Wi-Fi, Bluetooth, cellulaire, etc.

Étendre la facilité de navigation offerte en extérieur par le GPS à l'*indoor* a été un objectif majeur de la recherche dans la dernière décennie. La plupart des solutions proposées reposent sur le troisième type de capteurs, notamment par l'intermédiaire de l'infrastructure Wi-Fi [18, 54, 56, 62, 68, 80, 81]. Pourtant, la phase de prédéploiement pour la prise d'empreintes rend impossible la mise en place de telles solutions à large échelle. Pour éviter ce prédéploiement, d'autres solutions proposent d'utiliser des équipements spécifiques [44, 50, 82, 83], ce qui augmente le nombre de bâtiments où le système peut fonctionner mais réduit le nombre d'utilisateurs.

En théorie, Alice pourrait être capable de suivre sa trajectoire grâce à la centrale inertielle de son smartphone à partir du moment - et de l'endroit - où le dernier signal GPS est perdu. Cette approche, appelée aussi navigation à l'estime et utilisée par Christophe Colomb [84], a rapidement été reconnue comme impossible en raison de la nécessité d'intégrer les signaux d'accélération, qui sont beaucoup trop imprécis [70, 85, 86]. En utilisant le comptage de pas, certaines solutions sont parvenues à s'affranchir de l'imprécision des valeurs de l'accéléromètre pour n'observer que sa forme, ce qui nécessite une précision bien moindre. Elles maintiennent alors l'erreur à quelques dizaines de mètres, mais demandent une recalibration périodique, par action de l'utilisateur ou acquisition de signaux GPS [54, 70, 71, 87] du fait de leur incapacité à mesurer la taille exacte d'un pas de l'utilisateur.

2.1.3 L'idée d'Acrux et les défis à relever

Notre solution, Acrux - du nom d'une étoile polaire du Sud, l'équivalent pour la navigation de l'étoile polaire du Nord -, est le premier système qui, en reposant uniquement sur les capteurs de mouvement du smartphone, parvienne à fortement réduire la dérive de localisation, permettant une utilisation dans la plupart des bâtiments. Acrux y parvient grâce à deux idées majeures. La première provient de l'observation d'une autre branche de la recherche : les capteurs de mouvement sont imprécis mais permettent de déterminer l'activité d'un utilisateur [88]. On peut notamment savoir si un utilisateur se déplace, est immobile, se brosse les dents, etc. Par ailleurs, nous avons remarqué qu'il n'y a pas une seule façon de marcher. Il y a plusieurs mots pour identifier les différentes sortes de déplacement pédestre : le piétinement, la course, la marche rapide, etc. Ces mouvements sont connus dans le domaine de l'analyse physiologique et sont appelés **locomotions**

[77, 78]. Ainsi, les capteurs de mouvement pourraient permettre d'identifier le type de déplacement effectué par son utilisateur. Or, ce type de déplacement change également la vitesse de déplacement de l'utilisateur. La seconde idée réside justement dans le fait que le fait de compter les pas n'est pas un moyen efficace [70, 71, 89] pour estimer la distance parcourue par un individu. À la place, Acrux estime directement la vitesse de déplacement, beaucoup moins variable.

Suite à ces observations, nous avons résolu quatre problèmes majeurs afin d'obtenir un système de navigation efficace.

1. Les capteurs de mouvement semblent les plus adaptés pour déterminer les activités d'un utilisateur. Comment parvenir à identifier les locomotions en utilisant uniquement les capteurs de mouvement du smartphone, alors que la position du smartphone peut varier ?
2. Le suivi de la position de l'utilisateur doit pouvoir être mené en temps réel et être transparent vis-à-vis de l'utilisateur. Comment réaliser cette identification en temps réel, sans une consommation excessive de la batterie du smartphone ?
3. La vitesse et la direction de déplacement de l'utilisateur sont les deux éléments nécessaires à la navigation à l'estime. Comment estimer ces paramètres à partir du moyen de locomotion ?
4. La navigation à l'intérieur d'un bâtiment peut se faire sur plusieurs étages. Que se passe-t-il quand l'utilisateur se déplace dans les trois dimensions ?

2.1.4 Architecture de la solution

La figure 2.1 présente l'architecture de la solution Acrux. Afin de ne pas dépendre de l'infrastructure existante, Acrux se fonde uniquement sur les données des capteurs de smartphone. Plus précisément, Acrux utilise les capteurs de mouvement, particulièrement peu consommateurs en batterie [90]. Pour permettre la navigation en trois dimensions, Acrux utilise également le baromètre.

Quand Acrux détecte la perte du signal GPS - une information remontée par Android -, il considère la dernière position acquise comme le *fix*, c'est-à-dire le point de référence. À partir de cet instant, Acrux ne positionne pas le smartphone dans le référentiel géodésique du GPS, mais dans un référentiel terrestre, lié au bâtiment, avec pour origine le *fix* - qui correspond souvent à l'entrée du bâtiment -, comme le montre la figure 2.2b. Cela permet à l'utilisateur de suivre sa trajectoire dans le bâtiment. Étant donné que les valeurs brutes remontées par les capteurs du smartphone sont des mesures prises dans le référentiel du smartphone, présenté sur la figure 2.2a, Acrux inclut un bloc **Phone**

Attitude qui convertit directement les données des capteurs de mouvement du référentiel lié au smartphone dans le référentiel terrestre. Pour ce faire, AcruX utilise le capteur d'attitude inclus dans les smartphones. Celui-ci donne la direction des axes du smartphone par rapport au trièdre terrestre. Des algorithmes plus poussés peuvent permettre de réduire l'imprécision de ce changement de référentiel [91]. Cependant, comme AcruX ne repose pas sur les valeurs des capteurs mais sur les formes des signaux, un algorithme plus précis mais aussi plus complexe n'est pas nécessaire. À la sortie du bloc **Phone Attitude**, on obtient donc les signaux des capteurs de mouvement dans le référentiel terrestre lié au bâtiment.

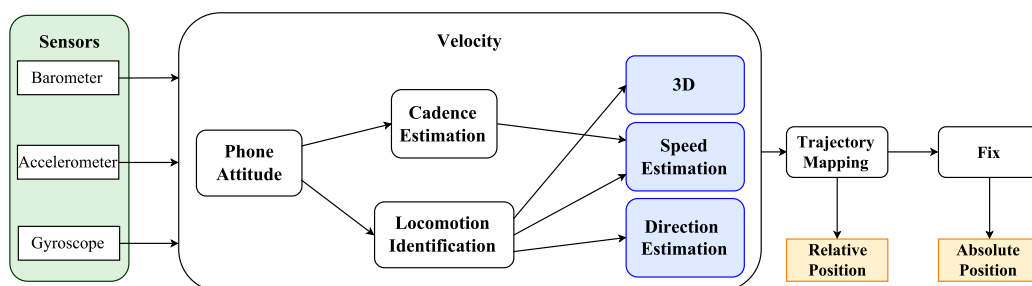


FIGURE 2.1 Architecture d'AcruX. AcruX utilise les capteurs de mouvement et un point de référence, appelé *fix*, pour calculer la position de l'utilisateur en temps réel. Pour cela, AcruX détermine la locomotion et la cadence de l'utilisateur.

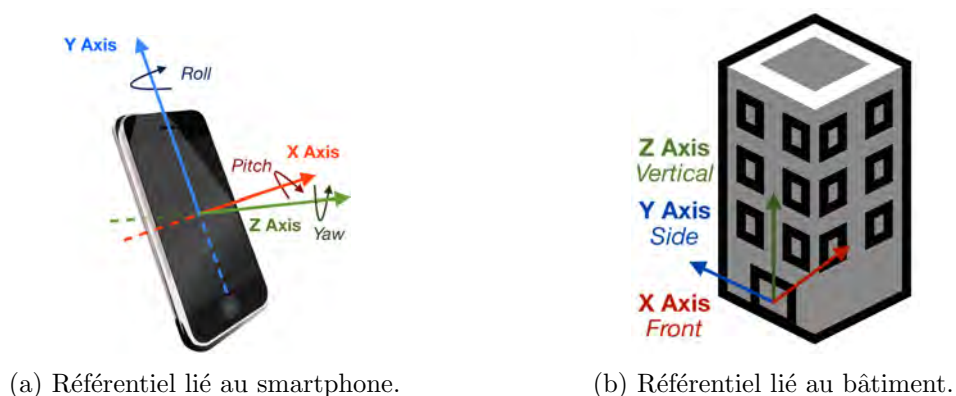


FIGURE 2.2 AcruX convertit les données du référentiel du smartphone vers celui du bâtiment.

Les remontées des capteurs fournies par le **Phone Attitude** permettent de définir d'une part la **locomotion de l'utilisateur** et d'autre part sa **cadence de pas**. Ces deux informations sont ensuite utilisées par les trois blocs principaux : le bloc **Speed Es-**

timation combine la cadence et la locomotion pour déterminer la vitesse de déplacement de l'utilisateur ; le bloc **Direction Estimation** combine les remontées du gyroscope et la locomotion pour déterminer la direction du déplacement ; le bloc **3D** permet de suivre l'utilisateur dans ces transitions entre les différents étages du bâtiment. Les transitions prises en compte sont les escaliers, les escalators (statique ou en montant) et les ascenseurs.

Grâce à la vitesse, la direction (en deux dimensions) et les mouvements verticaux estimés par les trois blocs en bleu, le trajet en trois dimensions de l'utilisateur peut être tracé. Si l'on connaît une position de l'utilisateur (le *fix*), on peut alors tracer cette trajectoire dans le référentiel du bâtiment, à partir de ce point connu.

Dans les sections suivantes, nous présentons tout d'abord notre algorithme d'identification du mouvement humain en six locomotions élémentaires. Nous montrons ensuite comment estimer la cadence de pas, la vitesse de l'utilisateur et la direction de déplacement. Nous proposons alors un algorithme pour le passage en trois dimensions. Enfin, nous menons une étude approfondie des algorithmes proposés, indépendamment et globalement.

2.2 Les locomotions humaines

En observant les déplacements d'un utilisateur de smartphone dans un bâtiment, nous avons séparé les mouvements en six **locomotions élémentaires**. Ces locomotions sont déjà identifiées dans le langage courant.

1. *Static* (statique) : l'utilisateur est immobile (assis par exemple).
2. *Idle* (à l'arrêt) : l'utilisateur peut effectuer des mouvements, mais ne se déplace pas (debout en train de bouger, le smartphone bouge, mais l'utilisateur ne change pas de position).
3. *Strolling* (piétinement, faire les cent pas) : l'utilisateur se déplace avec précaution, par exemple quand il réfléchit dans un bureau.
4. *Walking* (marche) : l'utilisateur marche normalement.
5. *Rushing* (marche à grandes enjambées, se presser) : l'utilisateur se déplace de façon décidé, par exemple quand il a une urgence.
6. *Running* (course) : l'utilisateur court, à l'intérieur d'un bâtiment.

Cette sélection de six locomotions élémentaires est un compromis entre la précision d'estimation et la capacité de les détecter en temps réel. Il s'agit des mouvements principaux que l'on peut observer dans un bâtiment. Ils sont par ailleurs étudiés dans le

domaine de la physiologie [92–95].

Dans cette section, nous présentons ce que sont ces locomotions humaines et leurs particularités, tant d’un point de vue physiologique que leurs conséquences sur les signaux des capteurs de mouvement. Nous identifions les locomotions élémentaires qui nous intéressent pour la navigation *indoor* et présentons un algorithme pour les détecter en temps réel. Enfin, nous mettons à l’épreuve notre algorithme pour analyser ces performances dans des cas réels.

2.2.1 Particularités des locomotions humaines

La figure 2.3 montre les signaux des trois capteurs de mouvement principaux pour identifier un pas humain. L’accélération horizontale est la plus intuitive dans la mesure où elle représente le mouvement vers l’avant de l’utilisateur. C’est le capteur utilisé dans la plupart des recherches existantes pour le comptage de pas [70, 71, 89, 96]. Nous avons ajouté à cela le capteur d’accélération verticale, étant donné que le sol provoque une réaction verticale sur l’utilisateur, et la vitesse de rotation autour de l’axe vertical (aussi appelée lacet) provenant du gyroscope puisque le corps (notamment les bras) bouge autour de l’axe vertical lors d’un pas. Les signaux de la figure 2.3 ont été collectés lors d’un déplacement de deux pas sur un HTC Google Nexus 5.

L’utilisateur tient son smartphone dans sa main droite et commence par un pas du pied droit. À t_0 , son pied droit quitte le sol. On peut apercevoir une augmentation de l’accélération frontale jusqu’à t_1 . À cet instant le pied droit continue son avancée jusqu’à atteindre le niveau du corps. L’accélération diminue alors jusqu’à ce que le pied heurte le sol à t_2 . Le sol transfère alors la vibration sur le corps, ce qui cause immédiatement deux pics dans les accélérations frontale et verticale (à t_3 et t_4). Dans le même temps, le bras droit tourne autour du corps de l’arrière vers l’avant. On peut le voir avec l’augmentation du lacet. Finalement, le pied s’applatit au sol, provoquant un léger pic dans l’accélération verticale (t_5).

L’utilisateur continue son mouvement avec un second pas, cette fois du pied gauche. Le même type de signal est observé pour les accélérations verticale et frontale. La puissance du signal est légèrement plus faible ce qui s’explique par le fait que l’utilisateur est droitier, et le pied droit dicte le mouvement. Cependant, on observe les mêmes pics aux mêmes moments. On peut également voir que, logiquement, le signal de lacet est inversé : la main droite tourne dans l’autre sens. C’est une première étape d’amélioration du comptage de pas : étant donné que deux pas avec la jambe droite, par exemple, ne vont pas s’enchaîner, on peut identifier un pas gauche qui aurait été oublié entre les deux,

uniquement en regardant le signal de lacet.

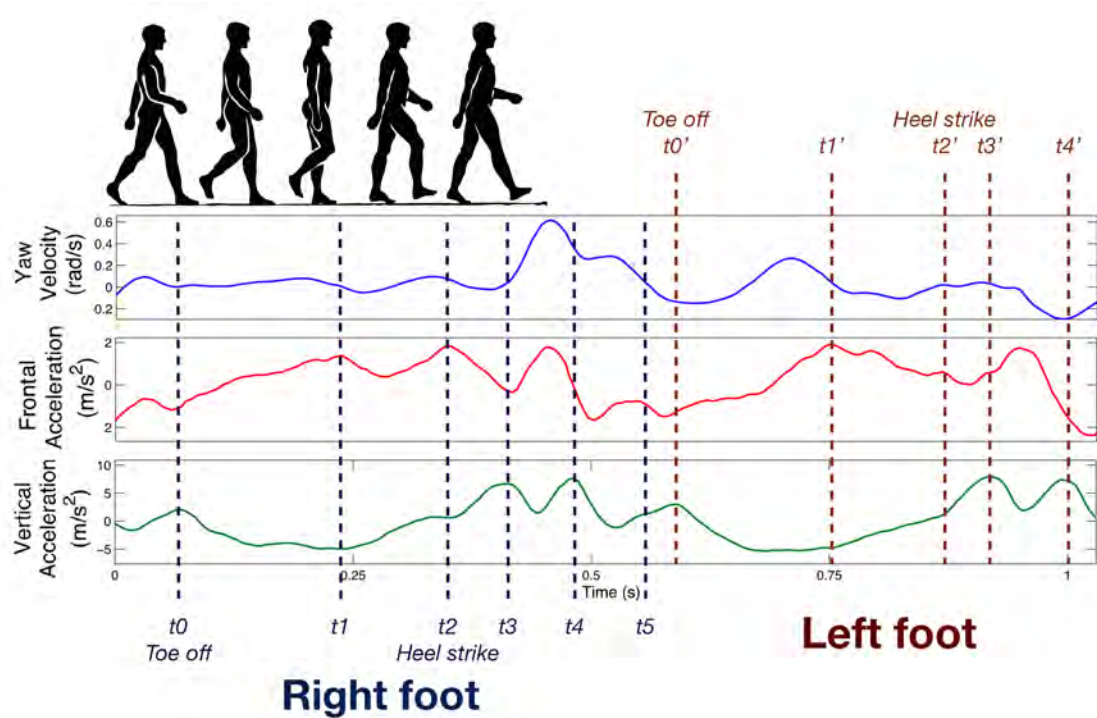
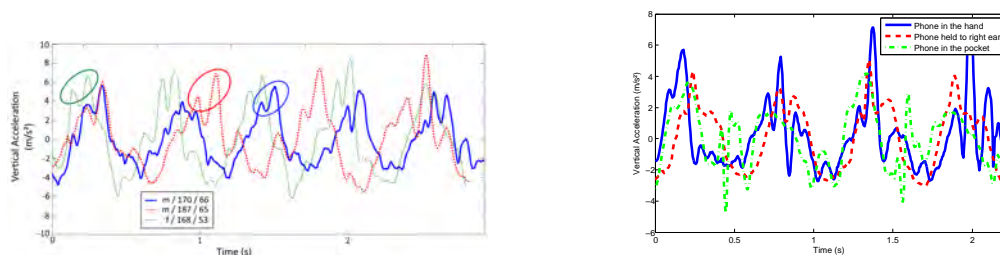


FIGURE 2.3 Signaux les plus importants dans la détection du pas humain pour un mouvement de deux pas.

Les figures 2.4a et 2.4b montrent par ailleurs, par le biais de l'accélération verticale par exemple, que le signal conserve une forme similaire pour trois utilisateurs différents et pour trois positions du smartphone également différentes. Dans le premier cas, ce sont trois utilisateurs (deux hommes et une femme) de corpulence différente dont on a observé le mouvement vertical. On peut notamment voir nettement les deux pics présentés précédemment lors du contact du talon sur le sol. Le même phénomène peut être observé dans le deuxième cas, où l'utilisateur a mis son smartphone dans différentes positions : dans sa main, à son oreille ou dans la poche de son pantalon. Les valeurs du signal sont différentes, mais la forme reste similaire.

Nous appellerons cette forme de signal, pour l'ensemble des six capteurs de mouvement, **signature** de la locomotion.



(a) Accélération verticale lors de cinq pas pour trois utilisateurs différents. (b) Accélération verticale lors de quatre pas pour trois positions de smartphone différentes.

FIGURE 2.4 Le signal d'accélération verticale reste similaire pour plusieurs positions de smartphone et différents utilisateurs.

2.2.2 Signature des locomotions

La figure 2.5 présente les signaux d'accélération verticale collecté à partir d'un HTC Google Nexus 5 pour les six locomotions sélectionnées. Les locomotions *static* et *idle* sont spécifiques : l'accélération est très faible et peu variable lorsque l'utilisateur ne bouge pas (statique), tandis que l'accélération est très variable, potentiellement avec de fortes accélérations lorsque l'utilisateur bouge sans se déplacer (*idle*). On peut facilement distinguer ce cas de celui des quatre locomotions suivantes où l'on observe un mouvement pseudo-périodique et répétitif. On peut également voir des différences dans les signaux de déplacement. Par exemple, le signal *strolling* (l'utilisateur piétine) ne possède pas les trois pics distinctifs du signal de marche.

La figure 2.6 présente en détail la distinction entre les signaux de marche et de course, à titre d'exemple. Les principaux signaux des capteurs de mouvement sont représentés pour quelques pas de marche et de course d'un même utilisateur. On peut voir notamment que juste après que le pied a heurté le sol, le signal d'accélération vertical est différent dans les deux cas. En effet, les deux pics ne sont pas du tout semblables pour la course, à l'inverse de la marche. Une raison qui peut expliquer cette différence réside dans le fait qu'en courant, la majorité [97] des gens retombent avec force sur le talon qui absorbe le premier choc avant de restituer l'énergie verticalement. En marchant en revanche, la pose du talon est un peu plus douce, ce qui explique la similitude dans les deux pics. Cette mécanique de locomotion ne dépend pas de la taille, du genre ou du poids d'un utilisateur, c'est ce qui permet de construire une signature indépendante de la personne.

Ainsi, nous pouvons remarquer que, visuellement, il est possible de détecter la locomotion d'un utilisateur en regardant les signaux des capteurs de mouvement de son smartphone sur au moins deux pas.

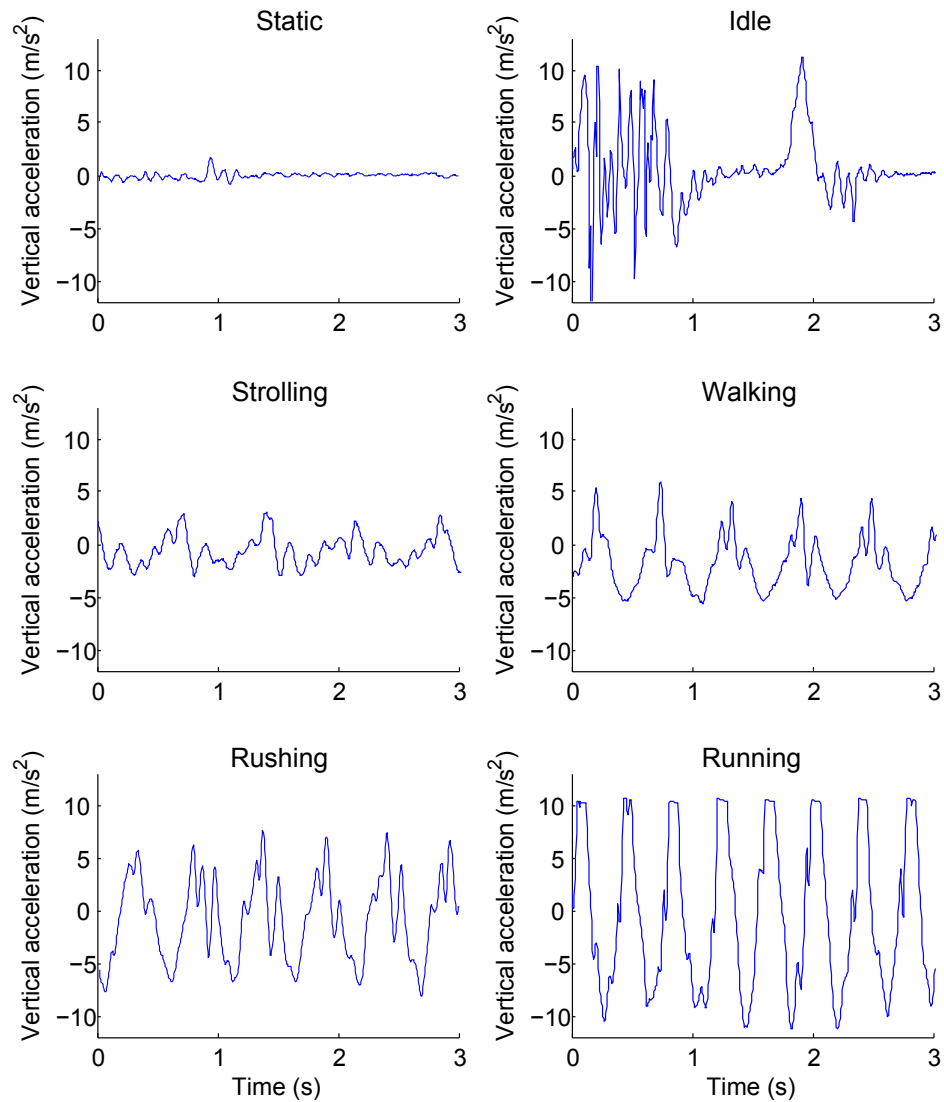
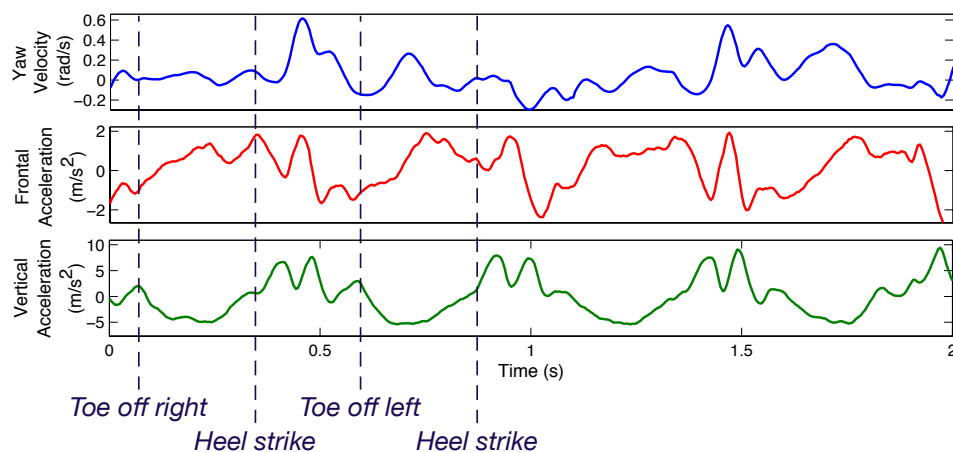


FIGURE 2.5 Accélération verticale pour les six locomotions élémentaires.

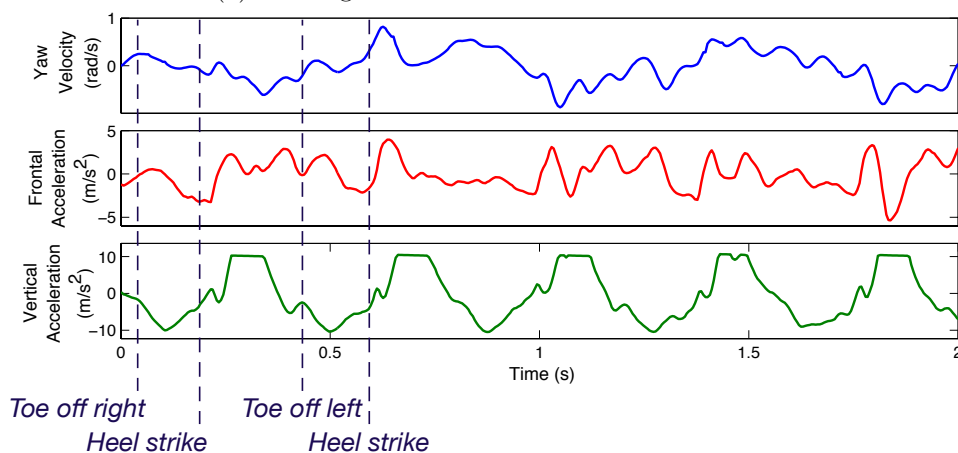
2.2.3 Détection de signatures

Comment automatiser cette détection ?

Cette analyse visuelle doit être complétée par un algorithme de détection automatique. AcruX enregistre la signature de chaque locomotion dans une base de données. Cette signature est commune à l'ensemble des utilisateurs et ne nécessite aucune action de leur part. Au moment de la navigation, l'algorithme compare les signaux des capteurs de mouvement enregistrés aux signatures stockées. La correspondance la plus proche



(a) Walking



(b) Running

FIGURE 2.6 Principaux signaux de capteurs pour deux locomotions différents.

correspond à la locomotion de l'utilisateur.

Création de la base de données des signatures

Plus précisément, nous enregistrons le signal de chaque capteur k pour chaque locomotion i . Soit $A_{i,k}(t)$ cet enregistrement, où $i \in \llbracket 1; m \rrbracket$ et $k \in \llbracket 1; p \rrbracket$. Ici, $m = 4$, pour les quatre locomotions hors *idle* et *static*, et $p = 6$: 3 capteurs pour l'accéléromètre et 3 pour le gyroscope. Ce signal est ensuite :

- tronqué, centré et complété avec du *zero-padding* ;
- converti dans le domaine fréquentiel $\mathcal{Z}_{i,k} = FFT(A_{i,k})$, et normalisé : $\mathcal{Z}_{i,k}|_{norm}$;
- pour chaque capteur k , j réalisations sont mixées pour obtenir une référence plus

- complète : $\mathcal{A}_{i,k} = \underset{\forall j}{Mix} (\mathcal{Z}_{i,j,k}|_{norm})$;
 — ajouté dans la matrice \mathcal{X} : $\mathcal{X} = (\mathcal{A}_{i,k})_{i \in \llbracket 1; m \rrbracket, k \in \llbracket 1; p \rrbracket}$.

Cette matrice \mathcal{X} est donc bien construite hors ligne et elle est commune à tous les utilisateurs.

Collecte des signaux en temps réel

En ligne, Acrux collecte le signal des capteurs de mouvement du smartphone et le compare à la matrice. Les signaux collectés subissent le même traitement que ceux utilisés pour la signature. Ils sont tronqués pour éliminer les effets de bord, centrés et complétés avec du zero-padding pour obtenir une taille de signal puissance de deux. Ils sont ensuite passés dans le domaine fréquentiel par transformée de Fourier. Chaque signal de capteur est comparé à son homologue stocké dans la matrice en utilisant le coefficient de Pearson. Pour deux signaux x et y de longueurs n , le coefficient de Pearson r est défini comme :

$$\begin{cases} r_{Pearson}(x, y) = \frac{xy^T - n\bar{x}\bar{y}}{\nu_x \nu_y} \\ \text{où : } \nu_x = \sqrt{\left[\sum_{i=1}^n x_i^2 \right] - n\bar{x}^2} \end{cases}$$

Détermination du poids de chaque capteur

Une fois les coefficients de Pearson calculés pour les six capteurs, nous calculons un score de corrélation scalaire en réalisant une moyenne pondérée des coefficients. Comme nous avons pu le voir visuellement, l'accélération verticale est le capteur le plus décisif dans la décision de locomotion. Il est donc nécessaire qu'il ait un poids plus important dans la décision que les autres capteurs. Pour identifier les capteurs les plus importants, nous avons calculé la corrélation \mathcal{W} entre tous les signaux d'un même capteur k pour une même activité i .

$$\mathcal{W}(k, i) = \frac{1}{J^*J} \sum_{(g,j)=(1,1)}^{(J^2)} r_{Pearson}(\mathcal{Z}_{i,j,k}, \mathcal{Z}_{i,g,k})$$

Nous avons alors accordé un poids autant plus important au capteur que la corrélation calculée est forte. Il en résulte que l'accélération verticale, conformément au constat visuel, est le capteur le plus corrélé, suivi de l'accélération frontale et du lacet.

Méthode de classification

Si aucune valeur d'accélération ne dépasse $2m/s^2$, la locomotion est *static*. Ce seuil a été identifié comme le maximum d'accélération obtenue sur plusieurs heures d'expérimentations statiques. Sinon, la locomotion choisie est celle qui obtient le meilleur

score. On obtient alors l'algorithme 1, qui permet d'atteindre un taux de détection de 91.2%.

Algorithm 1: Estimation du moyen de locomotion.

input : $1s$ de remontée des capteurs $(A(k))_{k \in \llbracket 1:p \rrbracket}$.

output : La locomotion estimée.

```

1 : begin
2 :    $p = 6$ ; //3 accéléromètres et 3 gyroscope
3 :    $m = 5$ ; //les 5 locomotions hors static
4 :    $W[6] = \{0.4, 0.2, 0.05, 0.2, 0.1, 0.05\}$ ; //poids des capteurs
5 :    $locomotion = \text{STATIC}$ ; //la locomotion par défaut
6 :    $score = 0$ ; //score à battre
7 :    $static\_thresh = 0.2$ ; //le maximum d'accélération pour la locomotion
   statique
8 :   for  $k = 1 \rightarrow p$  do
9 :      $z(k) = \text{frequency\_analysis}(A(k))$ ;

   //Identification de la locomotion élémentaire
10 :   if  $(\max(A(k)) < static\_thresh)$  then
11 :      $locomotion = \text{STATIC}$ ;
12 :   else
13 :     for  $i = 1 \rightarrow m$  do
14 :       for  $k = 1 \rightarrow p$  do
15 :          $r(i)(k) = r_{\text{Pearson}}(\mathcal{X}(i, k), z(k))$ ;
16 :          $score\_loc = r(i)^T \times W$ ;
17 :         if  $score\_loc > score$  then
18 :            $score = score\_loc$ ;
19 :            $locomotion = i$ ;
20 :   return  $locomotion$ ;

```

Taille de l'échantillon

Nous nous sommes posés la question de la taille d'échantillon idéale pour permettre la détection de la signature de locomotion en temps réel. Nous avons demandé à un utilisateur de se déplacer, en utilisant plusieurs locomotions, sur environ $200m$ en tenant toujours le HTC Google Nexus 5 dans sa main. Nous avons ensuite essayé de faire varier la fenêtre de prise de mesure entre $0.25s$ et $2.5s$. La figure 2.7 montre que plus l'échantillon est grand, meilleure est la précision. Cependant, cela augmente la fenêtre de

rafraîchissement. Nous avons choisi de conserver une taille d'échantillon de 1s, qui permet une précision de détection supérieure à 90% tout en conservant une taille d'échantillon et un temps de rafraîchissement raisonnables. Par ailleurs, cela correspond également au temps de rafraîchissement des scans Wi-Fi [18].

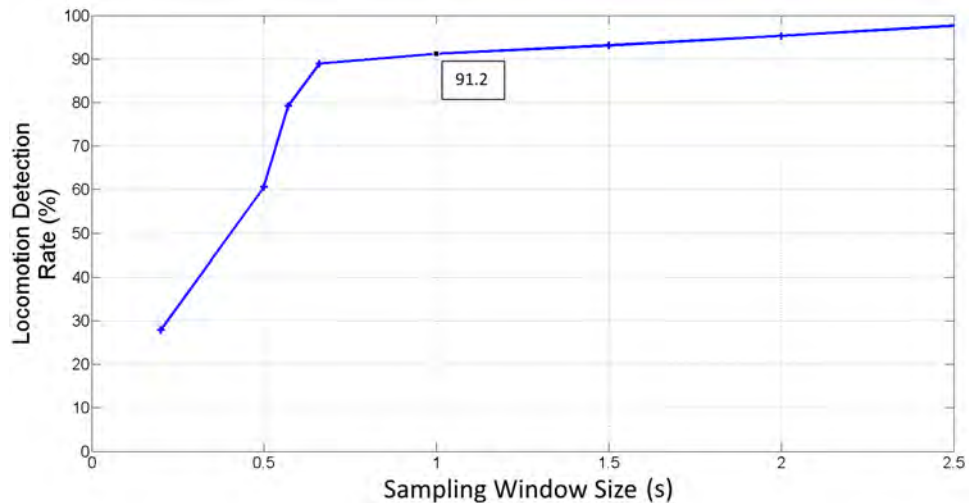


FIGURE 2.7 Taux de détection de la locomotion en fonction de la taille de l'échantillon.

Traitements en temps réel

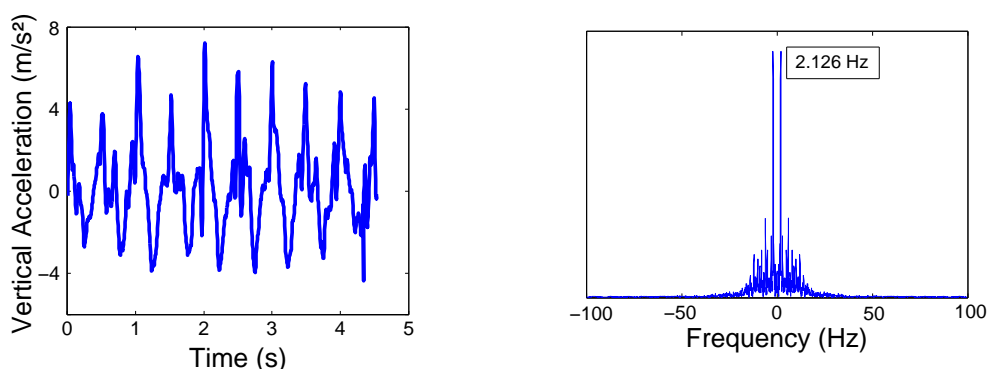
Ces traitements peuvent prendre du temps à être effectués. Or, sur un smartphone, il faut pouvoir effectuer ces tâches en temps réel, sans consommation excessive de mémoire et de batterie. Nous avons donc utilisé la transformée de Danielson-Lanczos [98] pour le passage dans le domaine fréquentiel, qui a une complexité en $O(N \log_2(N))$ pour des signaux de longueur puissance de deux. L'avantage de cet algorithme est que le signal calcule la transformée de Fourier en séparant deux processus pour la partie pair et impair du signal.

2.3 Vitesse et direction de déplacement

Dans cette section, nous présentons comment la locomotion, ajoutée au calcul de la cadence de pas peut permettre l'estimation de la vitesse de déplacement de l'utilisateur de smartphone. De la même façon, nous montrerons que la locomotion permet également de déterminer la direction de déplacement. La direction et la vitesse de déplacement permettent ainsi de suivre l'utilisateur à partir du point initial.

2.3.1 Mesure de la cadence des pas sur smartphone

Pour une locomotion donnée, on peut se déplacer à des vitesses différentes. En effet, même si la vitesse de piétinement n'atteindra jamais la vitesse de la course, on peut marcher ou courir à des vitesses différentes. L'élément le plus important dans ce changement de vitesse est la cadence de pas, c'est-à-dire le nombre de pas effectués chaque seconde. La figure 2.8 montre la relation directe entre la période du signal d'accélération verticale et la cadence de l'utilisateur. Il est donc possible de détecter la cadence de pas d'un utilisateur en observant le pic principal de la transformée de Fourier de son accélération verticale.



(a) Accélération verticale d'un utilisateur en *running* (b) Transformée de Fourier du signal d'accélération verticale

FIGURE 2.8 Le pic principal de la FFT permet de retrouver la période du signal : ici, c'est la cadence de pas.

2.3.2 Cadence, vitesse et locomotion

Connaissant la cadence de pas, nous utilisons une approche empirique pour déterminer la relation entre la cadence et la vitesse pour chaque locomotion élémentaire. Pour cela, nous avons donné le HTC Google Nexus 5 à plusieurs utilisateurs et nous leur avons demandé de se déplacer avec les quatre moyens de locomotion sur vingt mètres. Nous avons utilisé un chronomètre pour déterminer leur vitesse et calculer leur cadence. La figure 2.9 présente les résultats de cette expérimentation. Nous avons utilisé ensuite une régression polynomiale pour joindre les points et créer des courbes continues.

Nous obtenons ainsi la distance parcourue par l'utilisateur, en combinant cadence et locomotion pour obtenir sa vitesse, comme le montre l'algorithme 2.

La figure 2.10 montre le changement de vitesse lors d'un changement de locomotion.

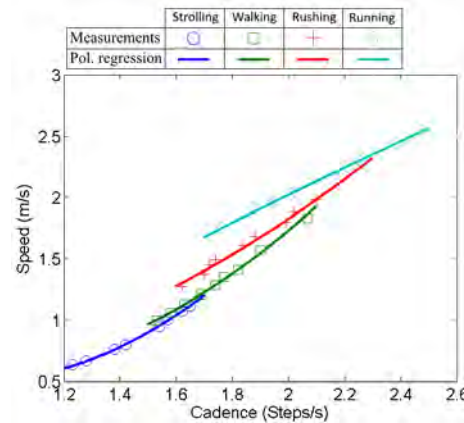


FIGURE 2.9 Vitesse en fonction de la cadence pour les quatre locomotions élémentaires.

Algorithm 2: Estimation du moyen de locomotion.

input : $1s$ de remontée d'accélération verticale ($A(a_z)$); la locomotion.

output : la vitesse.

```

1 : begin
2 :    $cadence = periode(A(a_z));$ 
3 :    $speed = linear\_regr(cadence, locomotion);$ 
4 :   return  $speed;$ 

```

Initialement, l'utilisateur se déplace en marchant, puis se dépêche. Ce n'est pas uniquement sa vitesse qui a changé, mais aussi sa façon de se déplacer : il est en locomotion *rushing*. L'algorithme s'adapte rapidement (en $1s$) et adapte la vitesse de marche. On peut facilement voir une période de transition dans la façon de se déplacer en observant l'accélération verticale qui a une période instable autour de $53s$. On retrouve cet effet dans le calcul de la vitesse avec une augmentation quasi-linéaire de l'estimation.

Nous avons finalement voulu vérifier la pertinence de notre étude, en comparant la précision d'estimation de la distance parcourue par rapport à une simple double-intégration de l'accélération. La figure 2.11 montre la fonction de répartition (*Cumulative Distribution Function - CDF*) de l'erreur de distance, calculée sur quatre cent soixante trajets *indoor* de dix utilisateurs différents et d'une longueur comprise entre $80m$ et $120m$. On peut voir qu'AcruX permet d'obtenir une erreur inférieure à $1m$ pour 90% des cas, contre seulement 13% pour la double-intégration.

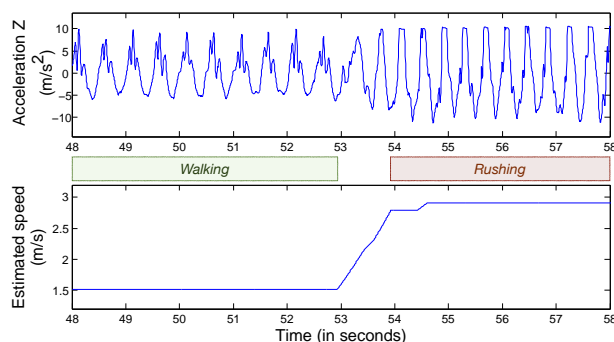


FIGURE 2.10 Estimation de la vitesse par Acrux lors d'un changement de locomotion.

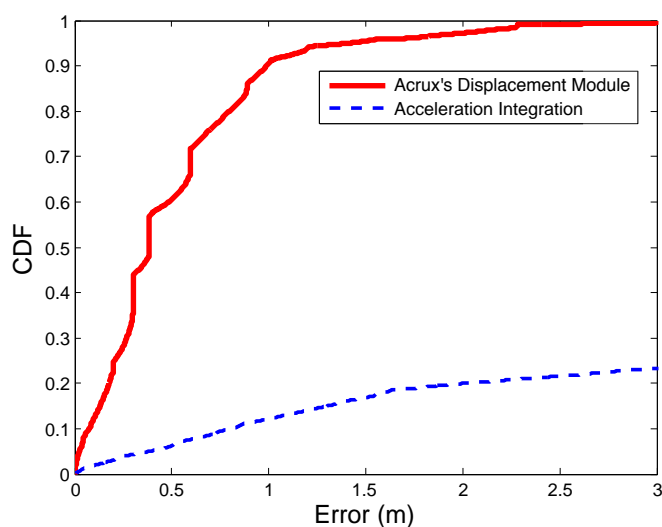


FIGURE 2.11 Fonction de répartition de l'erreur dans l'estimation de la distance parcourue pour Acrux et une double-intégration.

2.3.3 Direction de déplacement

L'estimation de la vitesse de déplacement doit se coupler avec celle de la direction de déplacement pour suivre l'utilisateur et obtenir sa position à chaque instant. En principe, le gyroscope MEMS (*microelectro-mechanical-system*) inclus dans le smartphone devrait être capable de suivre l'orientation du smartphone et donc de l'utilisateur. Pourtant, de la même façon que les accéléromètres, les smartphones embarquent un gyroscope à bas coût, dont les mesures sont imprécises. De plus, la direction d'un smartphone n'est pas directement liée à celle de l'utilisateur, puisque les mouvements parasites du déplacement humain brouillent les données du gyroscope, comme le montre la figure 2.12. L'utilisateur

marche droit puis tourne à gauche. On perçoit notamment un pic dans l'accélération latérale et dans le lacet. Cependant, on peut voir qu'il y a aussi des mouvements parasites, même sans changement de direction.

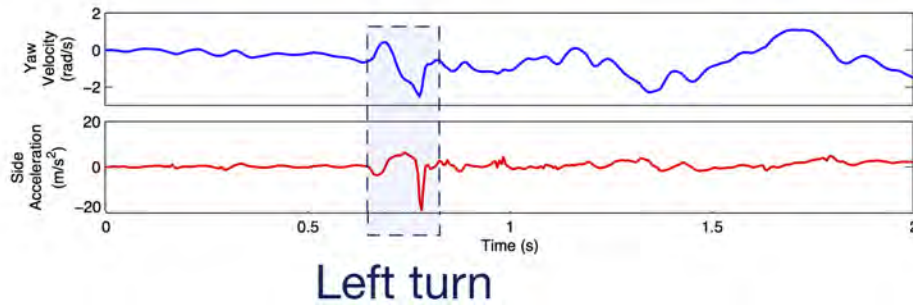


FIGURE 2.12 Lacet et accélération latérale pendant un virage à gauche.

Plusieurs solutions ont été proposées pour résoudre ces problèmes [74, 89], notamment en combinant les gyroscopes avec le compas inclus dans les smartphones. Heureusement, nous n'avons pas besoin de la direction de l'utilisateur à chaque instant, mais uniquement de sa direction globale de marche. De plus, il n'est pas nécessaire de connaître la direction dans l'absolu - c'est-à-dire par rapport au Nord par exemple -, mais uniquement dans le référentiel du bâtiment, qui est connu grâce au *fix* initial. La direction de déplacement estimée par Acrux repose donc sur deux idées principales : le filtrage des mouvements parasites, ainsi qu'une simple correction de biais, qui suppose la direction peu variable.

Filtrage par locomotion

Les remontées du gyroscope souffrant des mouvements parasites de l'utilisateur, Acrux utilise la signature de ces locomotions pour filtrer le signal de lacet obtenu du gyroscope. Pour cela, nous nous transposons dans le domaine fréquentiel dans lequel les mouvements parasites apparaissent pour la locomotion en cours (les signatures sont récupérées en ligne droite, et ne contiennent donc que des mouvements parasites pour l'estimation de la direction), et nous filtrons le signal courant remonté par le gyroscope. Nous obtenons ainsi une version filtrée du lacet qui donne une meilleure indication du changement de direction.

Correction du biais

Les gyroscopes possèdent un biais [99] qu'il est difficile de corriger *a priori*. Pourtant, dans notre cas, nous avons une information qui n'est pas négligeable : l'utilisateur qui utilise Acrux est en train de se déplacer. N'importe quel mouvement n'est donc pas

possible *a priori*. Par exemple, l'utilisateur ne va pas changer plusieurs fois brusquement de direction.

Soit $\dot{\theta}_m(t_n)$ la mesure filtrée du lacet à l'instant t_n . On peut supposer que la vitesse de rotation mesurée est la vitesse de rotation réelle $\dot{\theta}(t_n)$ à laquelle se rajoute le biais et un bruit, supposé blanc gaussien :

$$\dot{\theta}_m(t_n) = \dot{\theta}(t_n) + b(t_n) + \eta(t_n) \quad (2.1)$$

où $b(t_n)$ est le biais introduit par le gyroscope et $\eta(t_n)$ le bruit.

Dès lors, estimer la vitesse de rotation autour de l'axe vertical revient à estimer le biais à chaque instant. Nous pouvons écrire à chaque instant t_i :

$$\dot{\theta}_m(t_i) - \dot{\theta}(t_{i-1}) = \dot{\theta}(t_i) - \dot{\theta}(t_{i-1}) + b(t_i) + \eta(t_i)$$

Et en prenant l'espérance, nous obtenons par propriété de linéarité :

$$\mathbb{E} \left[\dot{\theta}_m(t_i) - \dot{\theta}(t_{i-1}) \right] = \mathbb{E} \left[\dot{\theta}(t_i) - \dot{\theta}(t_{i-1}) \right] + \mathbb{E} [b]$$

Si l'on ne suppose donc aucun changement brusque de direction, c'est-à-dire peu de variation de la vitesse de rotation en quelques millièmes de seconde et des mouvements dans les deux sens : $\mathbb{E} \left[\dot{\theta}(t_i) - \dot{\theta}(t_{i-1}) \right] = 0$, qui mène à :

$$\mathbb{E} [b] = \mathbb{E} \left[\dot{\theta}_m(t_i) - \dot{\theta}(t_{i-1}) \right] \quad (2.2)$$

On estime alors l'espérance du biais en utilisant la moyenne arithmétique :

$$\hat{\mathbb{E}} [b] = \frac{1}{n} \times \sum_{i=1}^n b(t_i).$$

Acrux peut estimer le biais à chaque instant t_n en connaissant les biais précédents et les valeurs mesurées du gyroscope :

$$\begin{aligned} \hat{b}(t_n) &= n \times \hat{\mathbb{E}} [b] - \sum_{i=1}^{n-1} \hat{b}(t_i) \\ &= n \times \hat{\mathbb{E}} \left[\dot{\theta}_m(t_i) - \dot{\theta}(t_{i-1}) \right] - \sum_{i=1}^{n-1} \hat{b}(t_i) \end{aligned} \quad (2.3)$$

Initialement, nous supposons que l'utilisateur marche dans une direction fixe, c'est-à-

dire : $\dot{\theta}(t_0) = 0$.

Performances de la détection

Pour tester la précision de la détection de direction d'Acrux, nous avons demandé à un utilisateur de suivre une trajectoire en forme de U en marchant avec le smartphone dans la main, afin d'estimer la capacité d'Acrux de suivre des chemins droits et deux angles à quatre-vingt dix degrés. La figure 2.13 présente les résultats obtenus dans trois cas : avec une simple intégration, avec uniquement le filtrage ou avec la correction de biais. On peut voir qu'après un peu plus d'un mètre, le mouvement parasite de l'utilisateur est filtré. La correction de biais, elle, se montre efficace dans les virages où l'angle estimé semble plus proche de la réalité.

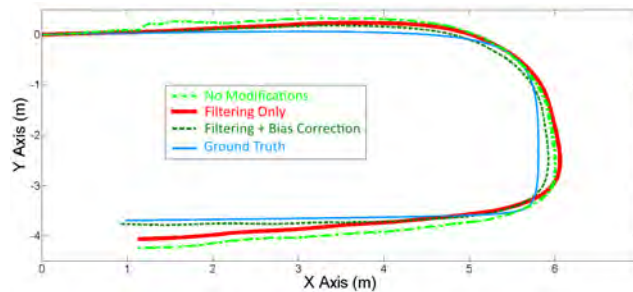


FIGURE 2.13 Trajectoires estimées par Acrux sans filtrage, avec filtrage et avec correction de biais pour un déplacement en U.

2.4 Modèle 3D

Dans cette section, nous décrivons comment nous avons étendu le fonctionnement d'Acrux à la 3D, c'est-à-dire aux déplacements de l'utilisateur dans les différents étages d'un bâtiment.

Nous présentons d'abord l'idée principale, qui est de reprendre la distinction en locomotion de la 2D, en séparant les activités de l'utilisateur lors de ses changements d'étage. En effet, pour changer d'étage, l'utilisateur va prendre l'ascenseur, l'escalier ou un escalator. La première étape consiste donc à détecter ces transitions spécifiques. Ensuite, nous montrons comment adapter l'algorithme 2D pour prendre en compte ces transitions.

Cette section présente donc les trois aspects majeurs permettant la navigation en 3D :

- l'estimation du changement d'altitude ;

- la détection du mode de transport vertical ;
- la détermination du déplacement de l'utilisateur entre les deux étages.

2.4.1 Estimation du changement d'altitude

Acrux calcule les changements d'altitude en utilisant le baromètre inclus dans les smartphones à partir de la formule suivante :

$$\Delta z = -\frac{\Delta p}{\rho g} \quad (2.4)$$

où Δz est la différence d'altitude, Δp est la différence de pression, ρ est la densité de l'air et g est la gravité terrestre (ou accélération de pesanteur).

Ce modèle simple suppose que la densité de l'atmosphère est uniforme. Pour déterminer la densité de l'air, nous supposons une température d'environ 21 °C. C'est une température commune à l'intérieur des bâtiments.

Le baromètre est, comme tous les capteurs d'un smartphone, imprécis. Surtout, il ne permet pas d'estimer l'altitude exacte de l'équipement. La figure 2.14 présente la variation du calcul d'altitude en fonction du temps pendant une heure, pour quatre smartphones différents posés sur une table. On peut voir que les remontées du baromètre sont extrêmement variables, la dérive pouvant atteindre jusqu'à 6m pour le HTC Google Nexus 4 et au minimum environ 2m pour le HTC Google Nexus 5. Par ailleurs, on peut voir qu'il y a de fortes différences entre les modèles de smartphone, probablement dues à la qualité variable des baromètres embarqués. Ces erreurs, de plusieurs mètres, montrent qu'il est impossible d'utiliser le baromètre pour connaître l'étage dans lequel se trouve un utilisateur : un étage fait moins de 4m de haut dans la plupart des bâtiments. Par ailleurs, la même conclusion est atteinte par [100, 101], qui proposent d'utiliser une empreinte de bâtiment pour permettre d'améliorer les résultats.

Cependant, notre objectif est de permettre la navigation d'un utilisateur dans un bâtiment. Ainsi, l'altitude exacte n'est pas une information nécessaire. En revanche, on cherche à déterminer l'étage dans lequel se trouve l'utilisateur, pour lui permettre de naviguer dans le bâtiment. En effet, de la même façon que pour la 2D, nous pouvons nous contenter de suivre les déplacements de l'utilisateur, et donc de ne déterminer que la variation d'altitude lors de son changement d'étage. Or, ce changement d'étage, quel qu'il soit, ne peut pas durer plus d'une minute. La figure 2.15 montre l'altitude déterminée par l'équation 2.4 sur un peu moins d'une minute, pour des utilisateurs statiques, prenant les escaliers et prenant l'ascenseur. La dérive et l'imprécision du baromètre est limitée sur cette échelle de temps, et on peut facilement distinguer le bruit du baromètre par

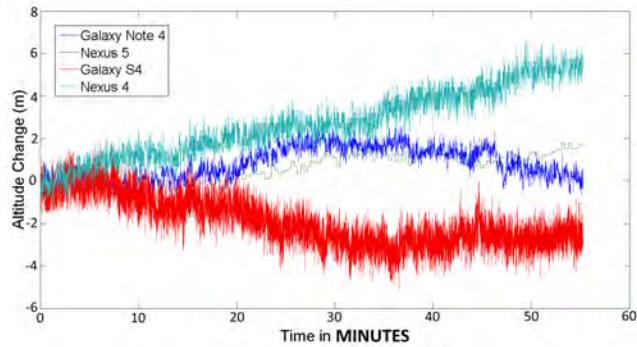


FIGURE 2.14 Altitude estimée par le baromètre pour des smartphones statiques pendant une heure.

rapport à des déplacements entre étages, que ce soit par escalier ou ascenseur.

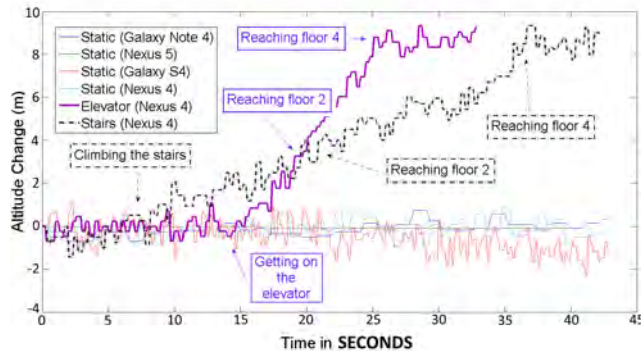


FIGURE 2.15 Altitude estimée par le baromètre pour plusieurs smartphones : quatre statiques, un dans un ascenseur et un montant quatre étages par des escaliers.

Ainsi, il est possible de détecter le nombre d'étages parcourus par l'utilisateur durant son déplacement vertical. Il reste à déterminer le mode de transport utilisé, afin de savoir s'il s'est aussi déplacé en deux dimensions.

2.4.2 Détection du mode de transport vertical

Tandis qu'un utilisateur empruntant un ascenseur ne bouge pas en 2D, celui prenant un escalator ou des escaliers se déplace. Il est donc nécessaire de connaître son type de transport. La figure 2.16 présente l'algorithme utilisé par Acrux. Dès qu'un changement d'altitude significatif est détecté ($> 1m$ en $5s$), le mode 3D est activé. Nous récupérons le mode de locomotion élémentaire calculé en deux dimensions. S'il s'agit de *static* ou *idle*, l'utilisateur est immobile. Si on a pu observé un pic très fort ($> 1m/s^2$) dans

l'accélération verticale, c'est que l'utilisateur a pris un ascenseur. Sinon, c'est qu'il monte un escalator. Si l'utilisateur se déplace en 2D, Acrux regarde la vitesse de changement d'altitude sur l'ensemble du trajet. Si l'utilisateur marche sur l'escalator, il est en plus propulsé par celui-ci, et sa vitesse verticale sera bien supérieure. Le seuil a été fixé à $0.5m/s$ par expérimentation.

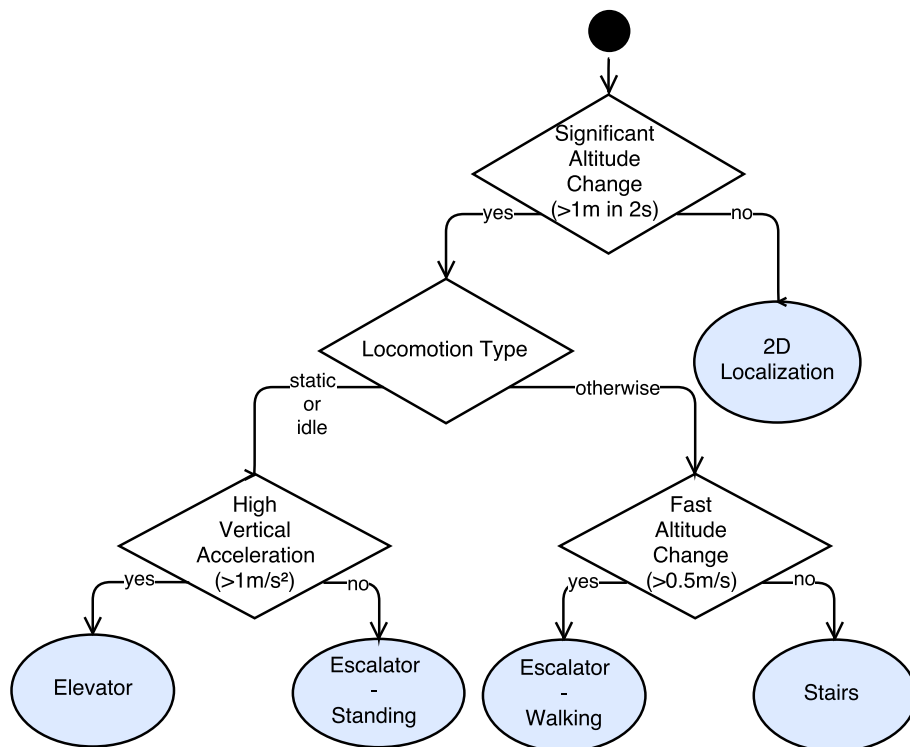


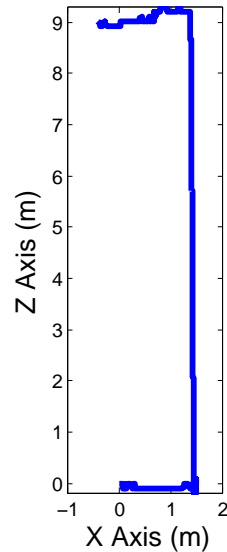
FIGURE 2.16 Algorithme pour la détection des activités 3D.

Étant capables de détecter ces quatre moyens de transport vertical, il nous reste à déterminer le déplacement 2D pour chacun d'entre eux.

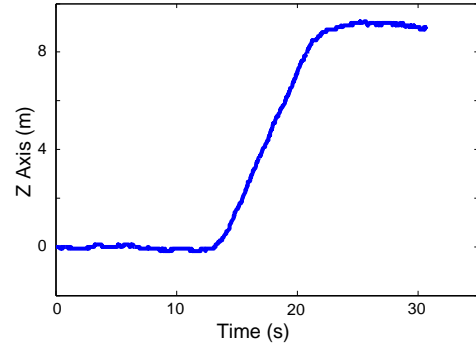
2.4.3 Ascenseurs

Pour déterminer le point d'arrivée de l'ascenseur, il suffit de mettre à jour l'altitude. Pour cela, Acrux calcule la différence d'altitude grâce à l'équation 2.4. La figure 2.17 montre les résultats obtenus pour un utilisateur qui prend l'ascenseur pour monter trois étages. On peut observer sur la figure 2.17a que l'utilisateur avance dans l'escalier puis reste immobile, monte de $3m$ avec l'ascenseur, puis sort par la même ouverture. La figure 2.17b montre l'évolution de l'altitude en fonction du temps. L'ascenseur a mis un peu

moins de 10s pour arriver à destination.



(a) Trajet d'un utilisateur prenant l'ascenseur.



(b) Déplacement vertical d'un utilisateur prenant l'ascenseur en fonction du temps.

FIGURE 2.17 Un utilisateur prend l'ascenseur et est suivi par Acrux.

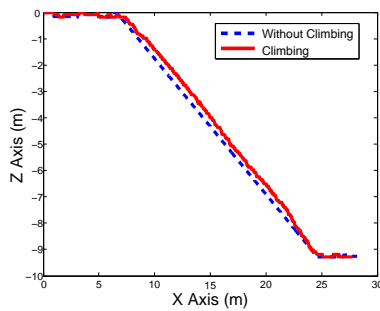
2.4.4 Escalators

Dans le cas des escalators, que l'on peut notamment retrouver dans les supermarchés ou les aéroports, l'utilisateur se déplace également sur le plan horizontal. L'angle d'inclinaison d'un escalator est réglementé ; il est de 30 degrés [102]. Grâce à cette information et une simple opération de trigonométrie, nous calculons la distance horizontale parcourue par l'utilisateur en fonction de son déplacement vertical estimé par le baromètre.

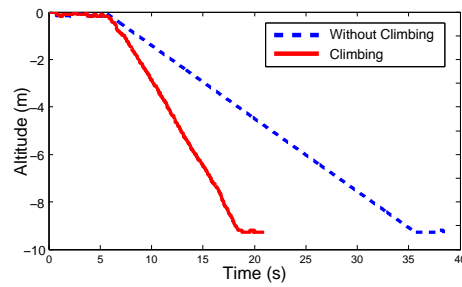
La figure 2.18 présente le suivi de deux utilisateurs avec Acrux, l'un descendant l'escalator et l'autre se laissant porter. On peut voir sur la figure 2.18a que les deux utilisateurs ont fait le même trajet. En revanche, sur la figure 2.18b, on observe que l'utilisateur ayant marché en descendant a atteint plus rapidement le bas de l'escalator.

2.4.5 Escaliers

Le cas le plus difficile est celui des escaliers. En effet, la forme des escaliers est très changeante : droit, en colimaçon, etc. Pour résoudre cela, Acrux utilise en parallèle du module 3D, l'estimation de direction pour répercuter le déplacement vertical dans le plan horizontal. Nous utilisons une fois de plus le fait que l'inclinaison des escaliers est également régulée, pour restreindre le nombre de chutes et d'accidents, autour de 30



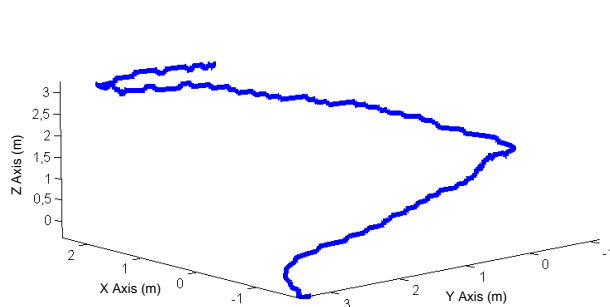
(a) Trajet des deux utilisateurs.



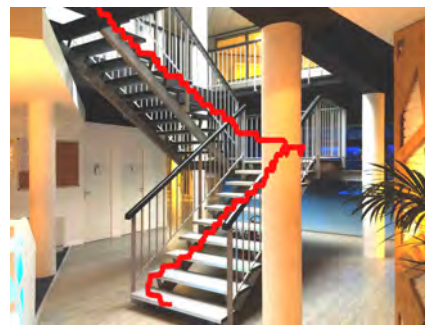
(b) Altitude des deux utilisateurs.

FIGURE 2.18 Deux utilisateurs prennent un escalator, l'un le descendant en marchant, l'autre restant statique lors de la descente.

degrés. La figure 2.19 montre les résultats obtenus par Acrux lorsqu'un utilisateur monte un escalier en U. On peut voir sur la figure 2.19b que le trajet superposé à une photo de l'escalier utilisé correspond bien.



(a) Trajet 3D de l'utilisateur dans l'escalier.



(b) Trajet estimé superposé à une photo de l'escalier.

FIGURE 2.19 Acrux suit un utilisateur prenant un escalier en U.

2.5 Performances d'Acrux

Pour évaluer Acrux, nous avons réalisé des expérimentations réelles avec dix utilisateurs, quatre smartphones et dans quatre bâtiments différents. Les smartphones testés sont : un HTC Google Nexus 5 et une tablette HTC Google Nexus 9 sous Android 5.0 Lollipop, un Samsung Galaxy S5 et un Sony Xperia E3 sous Android 4.4 Jelly Bean. Les quatre bâtiments sont des bureaux sur deux niveaux, une université sur cinq niveaux et deux stations de métro. Les dix utilisateurs ont des gabarits différents : sept hommes (entre $1.69m$ et $1.89m$, et entre $60kg$ et $72kg$) et trois femmes.

La plus grande difficulté dans l'évaluation d'Acrux a été de définir le trajet réel. En effet, les déplacements dans ces bâtiments privés et publics sont variés et il n'existe pas aujourd'hui de solution capable de connaître précisément la position d'un utilisateur, c'est même l'objet d'Acrux. Par conséquent, nous avons procédé ainsi. Pour chaque expérience, deux cobayes sont nécessaires pour chaque smartphone : l'un d'entre eux est l'utilisateur, porteur du smartphone ; l'autre est son ombre. Durant l'expérience, l'ombre suit l'utilisateur et marque le sol à chaque seconde. Lorsque le porteur a terminé son trajet, nous utilisons un mètre pour mesurer les coordonnées de chaque marque sur le sol. Cela correspond à ce qu'on appelle le trajet réel. Acrux calcule en parallèle la position dans le plan du bâtiment (le point de départ correspond à l'origine du référentiel). L'erreur de localisation est ensuite la distance euclidienne entre les deux coordonnées estimées.

Dans la suite de cette section, nous présentons donc les performances en 2D et en 3D d'Acrux, et nous présentons les perspectives de cette étude.

2.5.1 Performances *indoor*

Taille de pas vs Acrux

Idéalement, nous désirions comparer Acrux aux solutions complètes proposées dans la littérature, notamment les nombreux systèmes de comptage de pas [70, 71, 89, 96].

- [96] propose un modèle de la taille d'un pas en fonction des caractéristiques de l'individu (taille, poids, genre, etc.).
- [71, 89] proposent d'estimer la taille moyenne d'un pas à partir d'un modèle générique linéaire.
- [70] propose de la rétro-ingénierie pour mesurer la taille d'un pas. Pour un utilisateur donné, il compte le nombre de pas réalisés et le divise par la distance parcourue.

Malheureusement, aucun de ces systèmes n'est *open-source* et il est quasiment impossible de recoder soi-même ces solutions avec leur seule description dans les articles. Nous comparons donc notre solution à deux solutions génériques de comptage de pas :

- Taille de pas mesurée (*measured step size*) : nous mesurons la taille du pas moyen de l'utilisateur.
- Taille de pas groupée (*group step size*) : nous calculons la taille de pas du groupe d'utilisateurs.

Pour mesurer la taille de pas d'un individu, nous avons demandé aux dix volontaires de marcher normalement pendant 25m. La taille de pas moyenne est sélectionnée comme

étant leur taille de pas. La taille de pas groupée est la moyenne sur l'ensemble des dix volontaires.

La figure 2.20 représente les erreurs d'Acrux et des deux solutions de comptage de pas sur un trajet droit de 5km. Après 5km, Acrux dévie du trajet réel de seulement 8.33m soit 0.16% du trajet total, contre 52.29m pour le comptage de pas « idéal », ce qui représente une amélioration d'un facteur six. Cela est dû au fait que suivant sa locomotion et sa cadence, la taille de pas varie. Si on prend en compte une taille de pas moyenne, l'écart atteint 414.8m, ce qui est totalement inacceptable pour les solutions de navigation. Cependant, la majorité des trajets *indoor* sont beaucoup plus courts (de l'ordre de la centaine de mètres). Nous avons donc agrandi les premières centaines de mètres de l'expérience. Acrux n'atteint une erreur d'1m qu'au bout de 557m alors que le comptage de pas l'atteint au mieux au bout de seulement 99m.

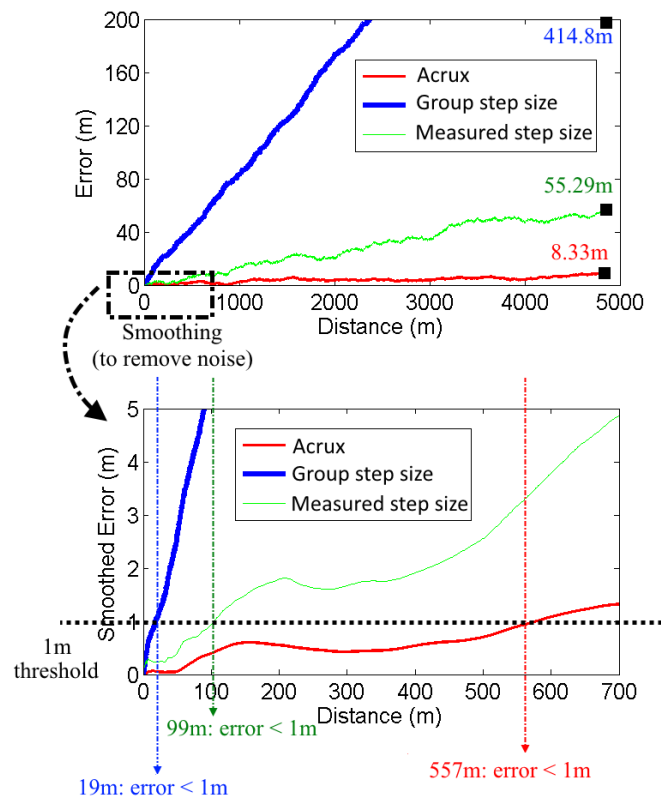


FIGURE 2.20 Erreurs d'Acrux et des solutions de comptage de pas en fonction de la distance parcourue sur un trajet de plusieurs kilomètres.

Acrux en *indoor*

Pour approfondir l'étude d'Acrux, nous avons sélectionné quarante-six trajets différents dans quatre bâtiments (trente-deux dans les bureaux, douze à l'université et deux dans deux stations de métro). Les dix utilisateurs se voient confier un smartphone et doivent se déplacer dans le bâtiment comme ils le font habituellement. Au total, la figure 2.21 présente la courbe cumulative d'erreur sur quatre cent soixante trajectoires *indoor* de longueurs comprises entre 80m et 120m. Acrux obtient une erreur médiane de 1.2m et un 98% percentile de 3m. Acrux permet donc de naviguer dans des trajets *indoor* en ne se reposant que sur les capteurs de mouvement et un unique *fix*.

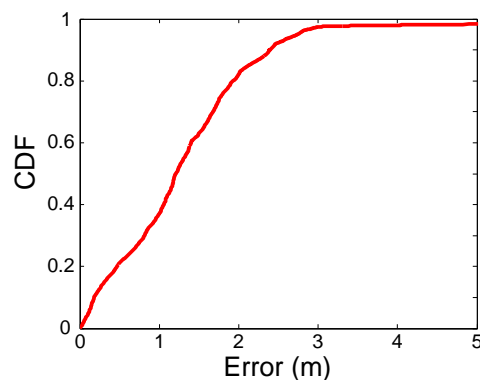


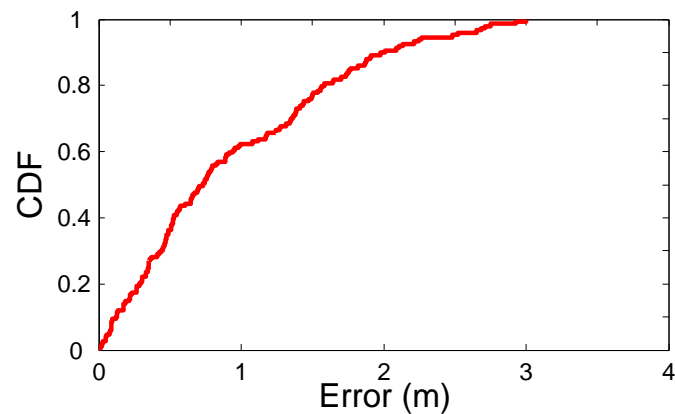
FIGURE 2.21 Fonction de répartition de l'erreur de localisation sur l'ensemble des 460 trajets de 80 à 120m.

Attrape-moi si tu peux !

Dans cette expérience, nous évaluons la capacité d'Acrux à suivre un utilisateur sur un trajet complexe à l'intérieur d'un bâtiment de bureaux. Un unique utilisateur est invité à suivre un trajet prédéfini présenté sur la figure 2.22b. Le trajet inclut plusieurs locomotions et plusieurs arrêts. Il inclut également des changements de direction dans les bureaux. La trajectoire totale est de 400m. Acrux parvient à suivre correctement la trajectoire. L'erreur médiane est de seulement 0.7m et n'excède jamais 3m comme le montre la figure 2.22a.

2.5.2 Performances *outdoor*

Afin de vérifier la précision et la validité d'Acrux, nous avons testé l'algorithme en extérieur, notamment pour comparer le trajet au GPS, qui est une référence plus fiable



(a) Fonction de répartition des erreurs de localisation d'Acrux par rapport au trajet réel, mesurées chaque seconde.



(b) Trajet imposé incluant plusieurs locomotions et plusieurs arrêts.

FIGURE 2.22 Acrux suit les mouvements de l'utilisateur dans un trajet *indoor* particulièrement difficile.

que nos trajets réels en intérieur. Un volontaire transporte un Samsung Galaxy S5 et un GPS Garmin 910 XT et suit le trajet de sa station de métro à son lieu de travail (plus de $700m$). La figure 2.23 présente trois trajets : le trajet réel, tracé à la main par l'utilisateur ; le trajet GPS capté par le Garmin 910 XT ; le trajet estimé par Acrux. Alors que le GPS démarre en extérieur, Acrux démarre sa détection à la sortie de la rame de métro. L'utilisateur prend un escalator pour rejoindre la sortie. Sur le trajet extérieur, l'utilisateur suit un tunnel pour passer sous un rond-point. Le GPS perd de la précision, tandis qu'Acrux continue de suivre l'utilisateur.

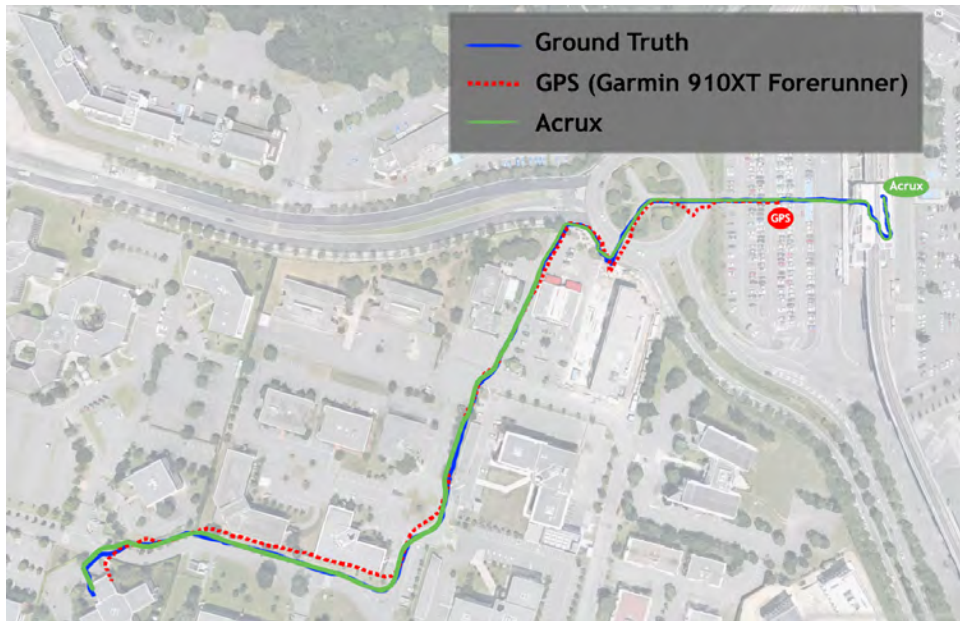


FIGURE 2.23 Acrux comparé au signal GPS sur un trajet classique : l'utilisateur part de sa station de métro préférée vers son lieu de travail.

2.5.3 Consommation de batterie

Nous avons également évalué l'impact d'Acrux sur la durée de vie de la batterie de l'utilisateur, élément non-négligeable pour les applications d'aujourd'hui. Pour cela, nous laissons Acrux en fonctionnement sur la tablette et les smartphones jusqu'à la décharge totale de la batterie. Dans le même temps, nous utilisons l'application Android App Power Tutor [103]. Nous comparons Acrux à BaselineRF une application simple, développée pour reproduire la consommation d'une application qui utiliserait le Wi-Fi pour localiser en *indoor*. Toutes les secondes, BaselineRF effectue un scan Wi-Fi passif. Le tableau 2.1 montre que la consommation d'Acrux est proche de celle d'une solution Wi-Fi. Acrux peut fonctionner pendant environ 14h sur tablette ou 5h du smartphone comme le montre le tableau 2.2. Cette faible consommation est due au fait que les capteurs de mouvement sont des capteurs à basse consommation énergétique [90].

2.5.4 Conclusions et perspectives

Dans ce chapitre, nous avons présenté Acrux, un système de navigation *indoor* qui utilise la navigation à l'estime grâce aux capteurs de mouvement du smartphone. Il est capable de conserver une précision de l'ordre du mètre sur des trajets de plusieurs centaines de mètres.

Application	Puissance (mW)	Batterie (Nexus 5)	Batterie (Tablette)
Acrux	452	5.09 h	14.8 h
BaselineRF	480	4.80 h	14.0 h

TABLE 2.1 Performance énergétique d'Acrux comparée à une solution classique de localisation de type Wi-Fi.

Équipements	Nexus 5	Galaxy S5	Xperia E3	Nexus 9
Batterie	2300mAh	2800mAh	6700mAh	2330mAh
Durée de vie	4.96 h	6.03 h	14.4 h	5.02 h

TABLE 2.2 Durée de vie de la batterie de plusieurs équipements utilisant Acrux.

Évidemment, il n'est pas possible de naviguer avec un seul point de référence à l'infini, et Acrux ne fonctionnerait pas dans n'importe quel scénario. Par exemple, Acrux dévierait beaucoup trop dans des tunnels de plusieurs dizaines de kilomètres comme à Toronto. Dans ces cas particuliers, Acrux pourrait être utilisé en combinaison avec un autre système qui permettrait : soit de corriger régulièrement les erreurs ; soit de fournir un positionnement précis à certains points du bâtiment, qui correspondraient à de nouveaux points de référence ou *fix*.

Acrux permet de tracer le trajet d'un utilisateur le long de son déplacement. Cependant, pour comprendre en quoi consiste le trajet dans n'importe quel bâtiment, le plan du bâtiment est nécessaire. La prochaine section propose une solution pour tracer le plan du bâtiment en même temps que le trajet de l'utilisateur.

UNE SOLUTION POUR CRÉER UNE CARTE DU BÂTIMENT DURANT LA NAVIGATION

Bob vient d'installer Acrux (ou une autre application de navigation sur smartphone) qui lui permet de connaître sa position et sa trajectoire dans un bâtiment. Il entre dans un bâtiment : un aéroport ou un grand bureau par exemple. Perdu, il lance Acrux. Sur son écran de smartphone, il peut observer chacun de ses mouvements et déplacements avec une précision de l'ordre du mètre. Pourtant, les plans de l'ensemble des bâtiments du monde ne sont pas connus par l'application. La trajectoire de l'utilisateur est donc tracée sur un fond vide. Bob ne sait donc pas où il va.

Dans ce chapitre, nous présentons InPreSS notre solution pour déterminer automatiquement le plan d'un bâtiment à l'aide des smartphones des utilisateurs d'Acrux. Ainsi, en plus de montrer à Bob sa trajectoire, InPreSS la trace avec, en fond, le plan du bâtiment, et lui propose un chemin pour atteindre sa destination. Par exemple, Bob peut connaître le chemin le plus rapide pour rejoindre sa porte d'embarquement dans un aéroport, se diriger vers le bureau de son collègue, ou encore aller à l'accueil d'un des nombreux services administratifs de la Préfecture. Pour cela, InPreSS étudie l'environnement de l'utilisateur via les capteurs d'ambiance du smartphone et détecte une dissimilarité lors d'un changement de pièce.

Ce chapitre est divisé en quatre sections. La première présente l'architecture de la solution InPreSS. En deuxième section, nous proposons l'algorithme d'identification de pièces à partir des capteurs du smartphone. Nous proposons alors de déterminer les chemins possibles entre les pièces d'un bâtiment afin de suggérer un trajet à l'utilisateur

pour rejoindre sa destination. Enfin, nous testons l'ensemble de la solution dans un bâtiment inconnu.

3.1 Système InPReSS

3.1.1 Historique de la construction de plans

Le défi de la construction d'un plan formé de chemins s'est d'abord posée en extérieur, pour les déplacements sur de longues distances. Dès l'empire romain [104], une carte des routes traversant l'Empire est construite par navigation à l'estime. Pour construire l'ensemble de la carte, les Romains ont dû parcourir tous les chemins possibles, suivre leur trajet petit à petit et regrouper l'ensemble de ces trajets sur une carte. Ce même principe est encore utilisé de nos jours : c'est notamment la solution choisie par Google Maps [25] ou Yahoo Maps [105]. Les plans sont obtenus à l'aide de voitures qui parcourent les routes en mesurant les distances et en prenant des photos. La création d'un plan à l'estime est beaucoup moins adaptée à un environnement intérieur. En extérieur, les routes sont identifiées et l'espace des chemins possibles est limité. En intérieur, comme on se déplace à pied, tout l'espace d'une pièce peut être utilisé. Afin de balayer l'ensemble du bâtiment, le domaine de la robotique s'est placé en tête pour résoudre le problème. À l'instar des robots ménagers, les robots parcourent tout le bâtiment par déplacements successifs. Dans le même temps, ils sont localisés (grâce à une centrale inertielle intégrée, appelée IMU-*Inertial Measurement Unit*) et cartographient l'environnement. C'est ce que l'on appelle le SLAM (*Simultaneous Localization And Mapping*) [30]. La limite de ce système réside simplement dans le fait qu'il est difficile d'imaginer envoyer des robots partout dans toutes les pièces. De plus, les robots peuvent être limités pour comprendre un environnement humain : les portes, la différence entre un objet obstruant le passage et un mur, etc.

La cartographie d'un espace complet a aussi fait l'objet d'études en extérieur. Google Maps utilise notamment les images satellites pour numériser le plan complet du sol. En imitant ce principe, JigSaw [106] propose d'utiliser les caméras des smartphones pour collecter des images de l'intérieur de tous les bâtiments. Au-delà du défi en termes de traitement d'images - les photos sont beaucoup moins stables avec des smartphones qu'avec des satellites -, cela impose un pré-déploiement particulièrement important : des utilisateurs doivent prendre des photos de toutes les pièces de tous les bâtiments. C'est inenvisageable à grande échelle.

Google Indoor Maps [24] a choisi une solution plus directe : ils ont demandé les plans des principaux aéroports et des plus grands centres commerciaux puis les ont

numérisés manuellement. Aujourd'hui, leur solution propose plus de 10000 plans. Ce nombre tend évidemment à croître mais ce n'est pas une solution définitive. D'une part, il est impossible d'envisager la collecte de l'ensemble des plans des bâtiments du monde entier. D'autre part, un environnement *indoor* est amené à être modifié : des portes peuvent être fermées, des murs construits, des pièces créées, etc. Chaque changement doit être accompagné d'une nouvelle collecte des plans du bâtiment.

3.1.2 La navigation à l'estime pour la création de plans

Historique

L'avènement des solutions de localisation *indoor* [3, 18, 54] et de *crowdsourcing* [55] a remis au goût du jour la création de plans à partir de la navigation à l'estime sur smartphone. En effet, l'effet de foule peut permettre d'envisager que l'ensemble des trajets des utilisateurs va couvrir l'intégralité du bâtiment. Plus précisément, ce type de solution ressemble au SLAM de la robotique. Or, la précision d'un système SLAM repose sur deux points essentiels.

1. La précision des mouvements du robot : sur smartphone, cela revient à la précision de suivi de l'utilisateur. Aujourd'hui, les solutions de navigation *indoor* sont limitées à quelques mètres [71]. De plus, les mouvements et trajets des utilisateurs sont imprévisibles, au contraire d'un robot.
2. La capacité du robot à détecter son environnement : les signaux captés par le smartphone, tels que le Wi-Fi, ne permettent pas de détecter les murs à eux seuls [107, 108].

WiFi-SLAM, graphSLAM [109] et [110] ont donc proposé d'utiliser une empreinte Wi-Fi du bâtiment, et notamment de profiter éventuellement de cette phase de *fingerprinting* hors-ligne parfois présente dans les solutions de localisation *indoor* [18]. Pourtant, cette phase est toujours lourde, et, comme on l'a vu dans les chapitres précédents, limite l'expansion de ces solutions. De plus, cela ne résout pas le problème de la mise-à-jour du plan du bâtiment lors d'un changement d'infrastructure.

Idéalement, une solution de construction de plans sur smartphone doit pouvoir les créer et les mettre à jour dynamiquement, sans avoir besoin d'une phase d'étude préalable de chaque bâtiment.

Objectifs

InPreSS a pour objectif de répondre aux exigences d'une application de navigation *indoor* en se fondant sur une solution de localisation existante. InPreSS peut, par exemple,

utiliser Acrux pour la localisation. Ainsi, s'appuyant sur une solution de localisation *indoor*, InPreSS cherche à proposer un système de reconstruction de plan pour la navigation dans le bâtiment. Lorsque l'on cherche à rejoindre une destination dans un bâtiment, il n'est pas nécessaire de connaître le plan exact. Il faut connaître les chemins qui permettent d'atteindre sa destination. Ainsi, nous chercherons plutôt à identifier l'ensemble des chemins entre les pièces du bâtiment qu'à reconstruire avec précision le plan de l'architecte.

3.1.3 Le principe d'InPreSS

Nous proposons donc InPreSS, une solution fondée sur les capteurs d'ambiance des smartphones et qui ne nécessite ni pré-déploiement ni action de l'utilisateur. InPreSS repose sur deux piliers :

1. la séparation du plan en pièces : pour cela, InPreSS estime l'homogénéité de l'environnement pour des points de mesure proches afin de détecter s'ils appartiennent à la même pièce ;
2. la détection des transitions entre les pièces : nous proposons un algorithme qui, à partir des déplacements des usagers entre les pièces, retrouve les transitions entre pièces et donc la présence d'une porte (ou d'une ouverture). Pour cela nous construisons un graphe.

Cadre applicatif

L'application fonctionne en trois temps.

1. Un premier utilisateur découvre le bâtiment : celui-ci ne se voit pas proposer de plan du bâtiment. En revanche, les données des capteurs de son smartphone sont récoltées tout le long de son chemin dans le bâtiment et envoyées au serveur InPreSS. InPreSS construit alors un plan à partir de ces données, stockées sur le serveur.
2. Éventuellement, comme dans les applications de navigation *outdoor* telles que [25], des utilisateurs peuvent indiquer la position de certains **points d'intérêt** sur ce plan.
3. Lorsqu'un nouvel utilisateur entre dans le bâtiment, il peut indiquer sa destination (en pointant un endroit sur le plan, ou en indiquant un point d'intérêt). L'application calcule alors un chemin vers la destination et le présente sur l'écran de l'utilisateur. Pendant ce temps, l'application collecte les données des capteurs de l'utilisateur pour mettre à jour le plan du bâtiment.

Point de vue système

InPreSS construit le plan d'un bâtiment en utilisant les capteurs d'ambiance des smartphones des utilisateurs qui s'y trouvent. Le seul élément externe nécessaire à InPreSS est la connaissance du contour extérieur du bâtiment. Celui-ci peut être obtenu par images satellites, ou directement dans des solutions existantes, comme [25]. L'architecture d'InPreSS est présentée sur la figure 3.1.

Quand un utilisateur entre dans un bâtiment, l'application collecte automatiquement les valeurs des capteurs d'ambiance de son smartphone. Ces valeurs sont localisées par un système de localisation indoor, tel qu'Acrux. Ces valeurs sont ponctuelles et le plan est continu. La première étape consiste donc à séparer le plan en zones représentées par un point.

L'étape suivante consiste à définir les zones qui appartiennent à la même pièce. Pour cela nous proposons d'utiliser les capteurs du smartphone. Deux méthodes sont alors possibles : détecter une homogénéité entre les points pour regrouper les zones en pièces ou mesurer une dissimilarité pour séparer des zones par des murs et ainsi former des pièces. Or, les zones identifiées ne sont pas indépendantes. Elles sont reliées par leur proximité. Deux zones non contiguës et éloignées ne peuvent pas appartenir à la même pièce. Par conséquent, plutôt que de regrouper les zones, il est plus simple de détecter la séparation potentielle entre deux zones : un mur. InPreSS propose donc, comme le montre la figure 3.1, d'estimer la distance (ou dissimilarité) entre les zones en fonction des remontées des capteurs, puis d'identifier les murs qui séparent les pièces.

InPreSS utilise ensuite l'ensemble des trajets des utilisateurs pour identifier les transitions entre les pièces : c'est-à-dire les portes ou les entrées qui permettent de passer d'une pièce à l'autre. La détection de ces transitions permet la construction d'un graphe représentant le plan du bâtiment, et permettant à l'application de proposer ultérieurement un trajet pour rejoindre n'importe quelle zone.

Point de vue utilisateur

Du point de vue de l'utilisateur, l'acquisition des données pour InPreSS est transparente. Il allume son application et entre sa destination. S'il est le premier à entrer dans le bâtiment, il servira de cobaye et InPreSS commencera son travail de cartographie. Sinon, InPreSS calcule le trajet optimal dans le bâtiment et le propose à l'utilisateur. Grâce à la solution de localisation, l'utilisateur est suivi pendant son trajet, comme le montre la figure 3.2. InPreSS en profitera pour continuer à enrichir ses informations.

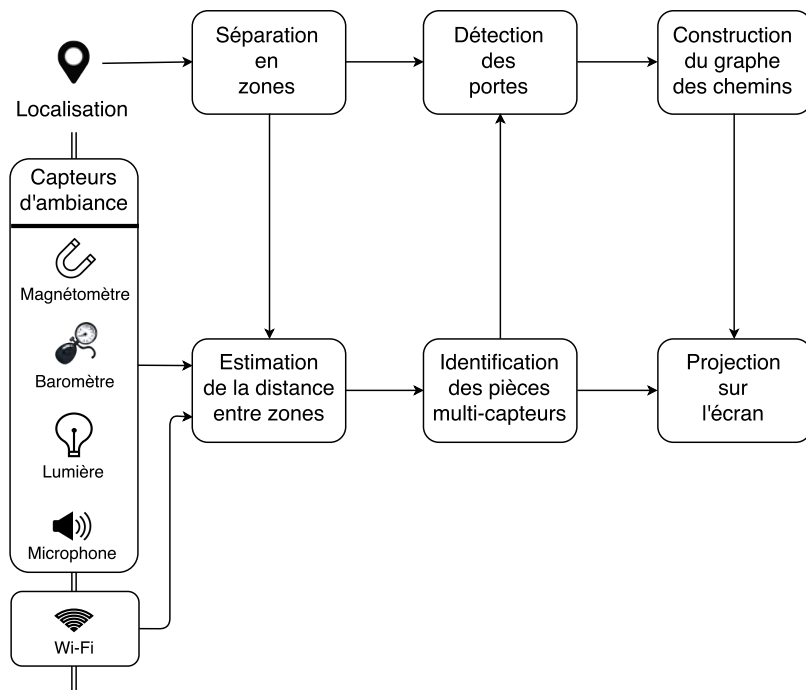


FIGURE 3.1 Architecture de l'application InPreSS.



FIGURE 3.2 Interface utilisateur de l'application InPreSS.

Fusion de deux plans établis par deux utilisateurs

Les données remontées par les capteurs de deux smartphones différents ou à deux moments de la journée ne correspondent pas obligatoirement. Il est impossible de comparer deux ambiances d'une même pièce selon qu'elle est sombre en soirée ou éclairée à midi. Deux capteurs distincts donnent également des résultats différents. On ne peut donc pas comparer deux points de mesure qui ne résultent pas d'un même utilisateur et d'un même trajet. InPreSS propose ainsi de ne pas comparer les deux familles de points de mesure, mais d'établir un plan fusionné à partir des plans obtenus par les trajets précédents.

Les sections suivantes présentent en détail les choix permettant de répondre aux défis posés par chacun des blocs de l'architecture InPreSS.

3.2 Identifier les pièces à l'aide des capteurs d'ambiance

Dans cette section, nous présentons l'approche d'InPreSS pour identifier les pièces d'un bâtiment. Pour cela, nous utilisons un bâtiment témoin de $480m^2$. Dans ce bâtiment, nous effectuons une prise de mesure dirigée, irréaliste dans un scénario réel où les utilisateurs peuvent se déplacer dans des directions différentes. Dans la dernière section de ce chapitre, nous testerons nos algorithmes dans un cas plus réaliste, où nous ne contrôlons pas les déplacements de l'utilisateur.

3.2.1 Séparation du bâtiment en cellules

Pour construire le plan du bâtiment, nous nous sommes appuyés sur une observation simple : quand Bob change de pièce, il ressent une différence d'environnement. Plus précisément, tous ses sens subissent une différence : la couleur, le niveau sonore, l'odeur ou la température sont différents d'une pièce à l'autre. Malheureusement, InPreSS ne peut pas reposer sur les sensations de Bob. Néanmoins, les smartphones modernes sont équipés de capteurs qui sont l'équivalent des cinq sens humains : le baromètre, le capteur de lumière, le magnétomètre, le microphone et le récepteur Wi-Fi. Nous avons donc eu l'intuition que ces cinq sens peuvent permettre au smartphone de ressentir le changement de pièce.

Pour vérifier cette intuition, nous avons mené une phase d'expérimentation dans des bureaux de $480m^2$ comprenant 11 pièces de tailles différentes (figure 3.3).

Un utilisateur, portant un LG Google Nexus 5, se promène dans toutes les pièces du bâtiment ; il ne s'agit pas d'un déplacement réaliste. Pour ce premier test, nous avons

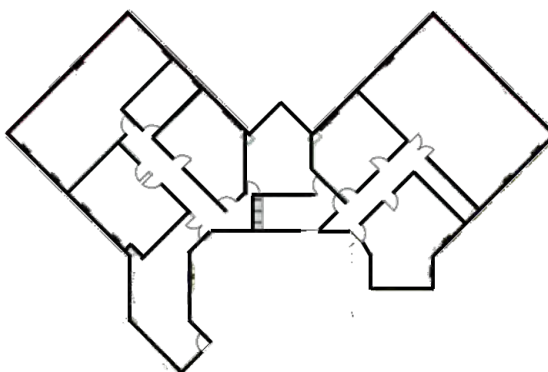


FIGURE 3.3 Plan du bâtiment expérimental.

imposé cela à l'utilisateur. Nous collectons les valeurs des cinq capteurs d'ambiance à intervalles réguliers. En tout, 80 points de mesure sont collectés afin de quadriller le bâtiment. Ils ont été répartis tous les deux mètres environ. Comme expliqué en introduction de ce chapitre, ces points de mesure sont pris lors d'un même trajet. Cela permet de ne pas subir les variations de l'ambiance d'une pièce au cours de la journée (variation de la lumière du jour par exemple). Sur un cas réel, les points seraient collectés régulièrement le long du trajet. Si plusieurs points étaient trop proches, ils pourraient être fusionnés. Nous n'avons pas codé cette partie dans cette version d'InPreSS.

Les points contiennent six informations :

1. la position ;
2. la pression ;
3. la luminosité ;
4. la puissance du champ magnétique - calculé à partir des trois axes du magnétomètre ;
5. le niveau sonore - calculé à partir des relevés du microphone ;
6. la puissance de signal Wi-Fi reçu : nous ne prenons que le signal maximum, de sorte qu'aucun scan spécifique n'est nécessaire.

Nous obtenons une base de données \mathcal{B} :

$$\mathcal{B} = [\mathcal{P}\mathcal{S}]$$

où

$$\mathcal{S} = (s_{i,k})_{k \in \llbracket 1;5 \rrbracket, i \in \llbracket 1;80 \rrbracket}$$

et

$$\mathcal{P} = (p_i)_{i \in \llbracket 1; 80 \rrbracket}$$

avec $p_i = (x_i, y_i)$ est la position de la mesure i , et $s_{i,k}$ est la valeur du capteur k à cet endroit.

Nous souhaitons alors colorer le plan suivant les valeurs des capteurs, afin de trouver d'éventuelles homogénéités entre les points d'une même pièce. Or, nous n'avons que des points, qui ne permettent pas de colorer une surface. Étant donné que nous ne connaissons pas la répartition des valeurs entre deux points de mesure, et que nous cherchons à identifier des frontières dans le plan, la solution d'une carte thermique n'est pas envisageable. Nous avons donc choisi de construire un diagramme de Voronoi [111]. Il permet de découper le plan en cellules, chacune représentée par son germe. Dans notre cas, les germes sont les 80 points de mesure. Tout point du bâtiment \mathcal{P} appartient à la cellule du germe dont il est le plus proche. On obtient alors 80 cellules de Voronoi $\mathbb{V}_{(i)}$ telles que :

$$\mathbb{V}_{(i)} = \{p = (x, y) \in \mathcal{P}, \forall j \in \llbracket 1; 80 \rrbracket, \|p - p_i\| \leq \|p - p_j\|\}$$

Nous avons choisi la norme euclidienne, de sorte que :

$$\|p - p_i\| = \sqrt{(y - y_i)^2 + (x - x_i)^2}$$

Nous obtenons donc une partition du plan du bâtiment, fondée sur le plan des murs extérieurs du bâtiment, comme le montre la figure 3.4.

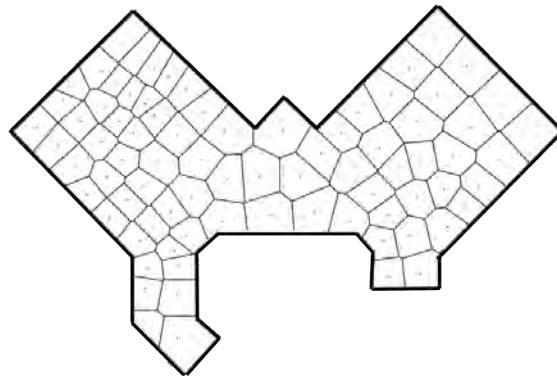


FIGURE 3.4 Partitionnement de Voronoi du bâtiment expérimental.

La figure 3.5 présente le niveau sonore calculé à partir du microphone au niveau des 80 germes. Chaque cellule de Voronoi se voit attribuer une couleur correspondant à la valeur de son germe. Plus le niveau sonore est élevé, plus la couleur est foncée. On peut remarquer que dans une même pièce, le niveau sonore - ou la couleur - est similaire. En revanche, il est impossible de retrouver les pièces avec un mécanisme à seuil : la grande salle à droite serait coupée en deux.

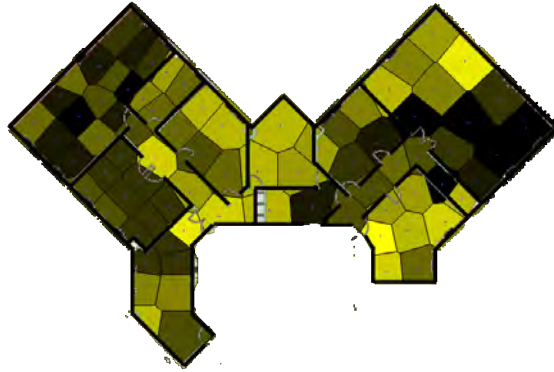


FIGURE 3.5 Niveau sonore dans le bâtiment.

De la même façon, la figure 3.6 présente la puissance du signal Wi-Fi reçue pour la cellule la plus proche. Plus la puissance est grande, plus la couleur est claire. Nous pouvons voir que, cette fois, la salle au fond à droite est homogène.

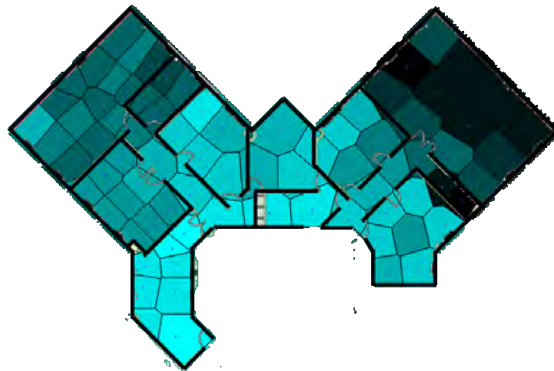
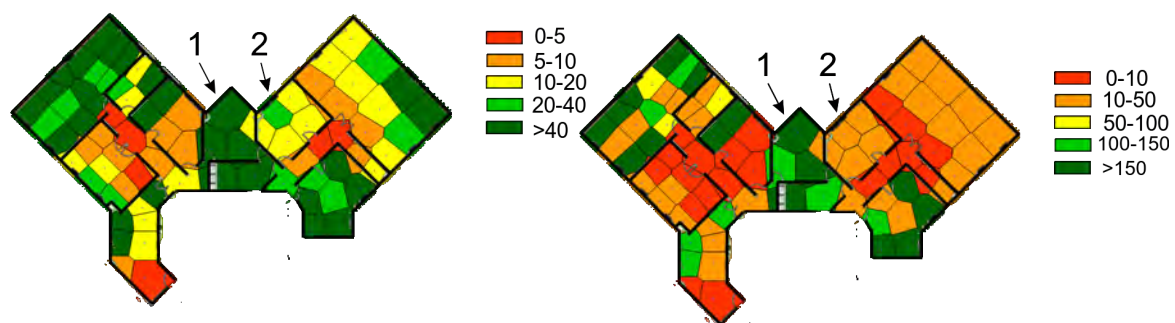


FIGURE 3.6 Puissance du signal Wi-Fi reçue dans le bâtiment.

Cette expérience conduit à deux constats principaux. Premièrement, il y a une homogénéité dans l'ambiance d'une pièce et les cinq sens d'un smartphone peuvent la percevoir. Un algorithme de détection de dissimilarité est donc nécessaire pour matérialiser la présence d'un mur séparant deux cellules. Par ailleurs, un seul capteur n'est pas capable d'une précision suffisante. En revanche, la combinaison des cinq sens du smartphone peut y parvenir. Les deux sous-sections suivantes visent à le vérifier.

3.2.2 Identification des murs à partir d'un capteur

Nous avons pu voir que les remontées de capteurs d'une même pièce sont similaires visuellement. Évidemment, ce n'est pas suffisant pour déterminer de façon automatique les frontières entre les zones similaires. La plus grande difficulté est donc de quantifier cette dissimilarité. Dans une carte colorée en fonction de la valeur du capteur, cela correspond à décider de la valeur des seuils entre les différentes couleurs. La figure 3.7 montre les plans colorés en fonction de la luminosité de chaque germe, pour deux niveaux de seuils différents. Sur la figure 3.7a la salle numéro 1 est parfaitement identifiée car elle est colorée entièrement de la même couleur (vert foncé). En revanche, la salle numéro 2 est identifiée uniquement pour des seuils plus élevés, comme le montre la figure 3.7b.



(a) Carte colorée de la luminosité pour des seuils bas. (b) Carte colorée de la luminosité pour des seuils hauts.

FIGURE 3.7 Cartes colorées de la luminosité dans notre bâtiment témoin pour deux niveaux de seuils différents.

Pour contourner cette sensibilité aux seuils, nous avons étudié la distribution des valeurs des différents capteurs. La figure 3.8 représente la fonction de répartition de la luminosité mesurée sur les 80 germes. La luminosité est concentrée sur des valeurs basses. Cela correspond aux lieux où il n'y a aucune forte source de lumière directe. Seules

quelques valeurs sont élevées. Ces valeurs sont celles qui sont captées juste en-dessous d'une lampe - en l'occurrence, d'un néon. Ainsi, si la dissimilarité entre deux germes est calculée par la valeur absolue de la distance entre les deux valeurs associées, cela peut conduire à des erreurs dans la séparation des pièces. Par exemple, deux germes d'une même pièce, l'un directement en-dessous du néon et l'autre juste à côté, auront deux luminosités très fortes mais avec une différence relative significative. Inversement, entre une pièce sombre et une pièce mal éclairée, la luminosité est faible et la différence relative aussi. La mesure de la dissimilarité par différence relative mènera à la séparation en deux de la première pièce et au regroupement des deux dernières.

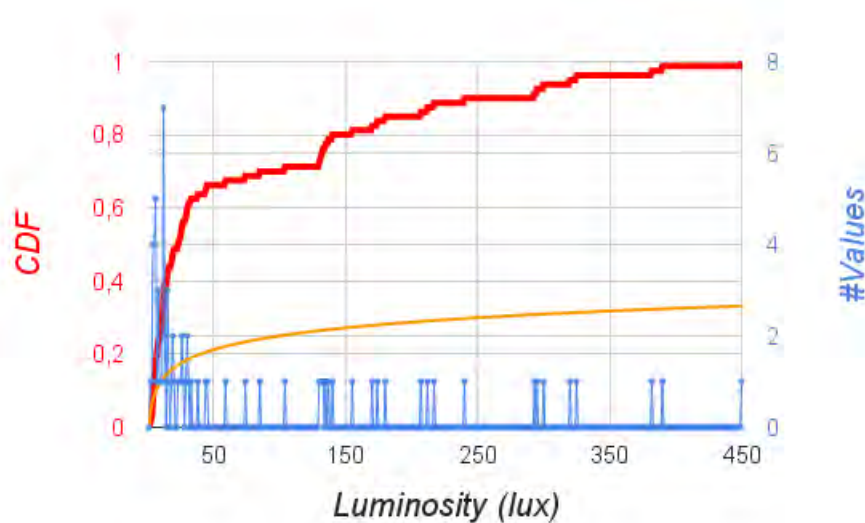


FIGURE 3.8 Fonction de répartition de la luminosité dans le bâtiment témoin.

Pour éviter de telles erreurs, il faut que la distribution des valeurs mesurées soit aplatie, de sorte que la différence entre les fortes valeurs soit moins influente que la différence entre de faibles valeurs. Dans le cas de la luminosité, cela correspond à une échelle logarithmique. La figure 3.9 présente la fonction de répartition du logarithme de la luminosité. L'effet obtenu correspond aux attentes : la fonction de répartition est pseudo-linéarisée [112].

La figure 3.10 montre une carte colorée du bâtiment, en utilisant cette fois l'échelle logarithmique. Les pièces 1 et 2 notamment sont correctement identifiées.

En observant la fonction de répartition de l'ensemble des valeurs provenant des capteurs, nous avons alors proposé la mesure d'une distance ou d'une dissimilarité pour chaque capteur, comme présenté dans le tableau 3.1. La pression, le champ magnétique et

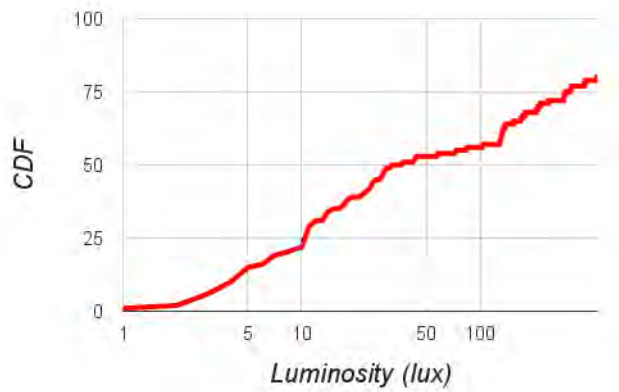


FIGURE 3.9 Fonction de répartition de la luminosité en échelle logarithmique pour l'abscisse.



FIGURE 3.10 Carte colorée de la luminosité en utilisant une échelle logarithmique.

le niveau Wi-Fi sont très variables et ne possèdent pas de valeur dominante. Nous avons donc choisi d'utiliser la simple norme euclidienne. En revanche, la distance en racine carrée est utilisée pour le niveau sonore, le niveau sonore est en effet plutôt concentré sur les faibles valeurs - les pièces sont plus souvent calmes que bruyantes.

3.2.3 Mise en commun de plusieurs capteurs

Un seul capteur n'est toujours pas suffisant pour distinguer parfaitement les pièces d'un bâtiment, comme le montre la figure 3.10. Afin de combiner l'apport des cinq capteurs d'ambiance des smartphones, une solution peut consister à utiliser un algorithme

Capteur	Nom	Distance $d_k(i, j)$
Pression atmosphérique	Euclidienne	$ s_{(i,1)} - s_{(j,1)} $
Luminosité	Logarithmique	$ \log(s_{(i,2)}/s_{(j,2)}) $
Champ magnétique	Euclidienne	$ s_{(i,3)} - s_{(j,3)} $
Niveau sonore	Racine carrée	$ \sqrt{s_{(i,4)}} - \sqrt{s_{(j,4)}} $
RSS Wi-Fi	Euclidienne	$ s_{(i,5)} - s_{(j,5)} $

TABLE 3.1 Distance utilisée pour chaque capteur.

de partitionnement en k -moyennes (*k-means* [113]), une classification par regroupement hiérarchique [114], ou encore un algorithme d'espérance-maximisation (EM [115]).

K-means permet de partitionner les 80 germes en k groupes homogènes - en termes de valeurs des capteurs. Il est nécessaire de connaître à l'avance le nombre k de salles. C'est évidemment une donnée que l'on n'a pas et qu'il est difficile d'estimer : chaque bâtiment à un nombre et une densité de pièces très différents. Le même problème existe pour l'algorithme EM.

La classification par regroupement hiérarchique permet de s'affranchir facilement de la connaissance du nombre de groupes. Initialement, toutes les graines sont séparées. Deux d'entre elles, les plus ressemblantes, sont fusionnées pour appartenir à un même groupe. Ce procédé est répété jusqu'à un critère de fin de l'algorithme. Ce critère peut être le nombre de groupes désirés, mais pas seulement : il peut aussi être un seuil de dissimilarité entre les groupes à ne pas dépasser. Dans notre cas, le principal inconvénient de cette méthode est qu'elle ne prend pas en compte une information supplémentaire : en plus des valeurs des capteurs, nous possédons également la localisation des germes. Ainsi, deux germes très ressemblants mais très éloignés ne peuvent pas appartenir au même groupe.

Inspiré par le regroupement hiérarchique, nous avons donc décidé d'utiliser la dissimilarité entre deux germes, tout en prenant en compte les zones de contact entre les deux cellules de Voronoi qu'elles représentent. Pour cela, pour chaque couple de cellules de Voronoi en contact, nous calculons la dissimilarité entre les deux mesures. Si cette dissimilarité est supérieure à un seuil, nous supposons qu'il existe une séparation - c'est-à-dire un mur sur le plan - à la frontière des deux cellules.

Nous proposons le coefficient de dissimilarité $D(i, j)$ entre la cellule i et la cellule j :

$$D(i, j) = \sum_{k=1}^5 \delta_k(i, j)$$

où $\delta_k(i, j)$ est la distance normalisée $d_k(i, j)$ en utilisant le capteur k .

$$\delta_k(i, j) = \frac{d_k(i, j)}{\sum_{l, n} d_k(l, n)}$$

Il s'agit de la distance euclidienne à cinq dimensions entre les cellules i et j reposant sur les distances $d_k(i, j)$ proposées précédemment. Nous aurions pu pondérer les différents critères. Cependant, l'optimisation de ces poids relatifs aurait été difficile à mener et risquait de dépendre du bâtiment.

L'étape suivante va déterminer le meilleur seuil. Pour cela, nous avons conduit une session de mesures dans deux autres bâtiments, de sorte à obtenir 100 cellules voisines : 50 d'entre elles sont dans la même pièce, alors que les 50 autres sont de part et d'autre d'un mur. Le meilleur seuil est celui qui minimisera le nombre de cellules de la même pièce séparées par l'algorithme (probabilité de fausse alarme), tout en maximisant le nombre de cellules correctement séparés (probabilité de détection). La figure 3.11 présente la fonction d'efficacité du récepteur (*ROC curve* pour *receiver operating characteristic*) qui trace la probabilité de détection en fonction de la probabilité de fausse alarme pour différentes valeurs du seuil. Pour la suite de notre étude, nous avons choisi le point rouge correspondant à un seuil de 0,8 qui permet d'atteindre une probabilité de détection de 84% tout en restreignant la fausse alarme à 10%. Cela permet de limiter le nombre de fausses séparations entre pièces à 10% des cas, ce qui semble un chiffre raisonnable par rapport aux techniques de type SLAM [30]

3.2.4 Performance du système de détection de pièces

Nous avons évalué les performances de notre système dans notre bâtiment témoin. La figure 3.12 présente les plans obtenus en utilisant deux méthodes de classification : k-means et le partitionnement hiérarchique, ainsi que notre solution InPreSS. La figure 3.12a a été obtenue en utilisant k-means avec précisément 12 pièces, c'est-à-dire en supposant que l'on a correctement estimé le nombre de pièces. Ensuite, nous attribuons à chaque groupe une couleur. Nous pouvons voir que l'algorithme échoue à identifier les

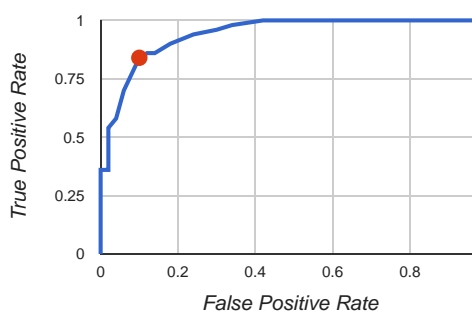
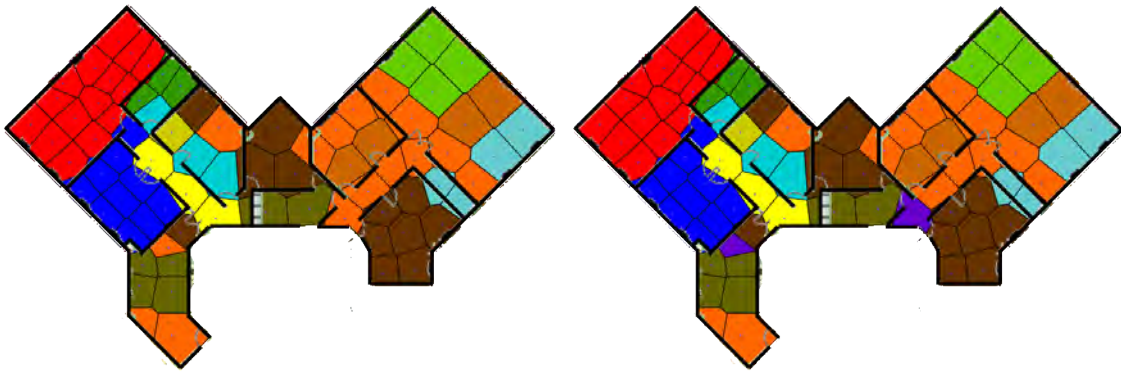


FIGURE 3.11 Fonction d'efficacité du récepteur selon les valeurs du seuil. Le point rouge correspond à un seuil de 0,8.

pièces. C'est encore pire si le nombre de pièces n'est pas correctement estimé, comme le montre la figure 3.12b. Le résultat obtenu avec le partitionnement hiérarchique est meilleur, comme le montre la figure 3.12c mais ne retrouve pas le plan du bâtiment, notamment car il ne prend pas en compte la position des germes. En revanche, la figure 3.12d montre qu'InPReSS parvient à retrouver la forme du bâtiment. Au contraire des méthodes de partitionnement classique, il ne regroupe pas les zones en groupe, mais sépare les cellules adjacentes dissemblables par un mur. Onze pièces peuvent être correctement distinguées. Il existe cependant une différence avec le plan réel du bâtiment : certaines pièces ne sont pas complètement fermées. En effet, si la porte est ouverte, les deux pièces qu'elle sépare peuvent se ressembler à l'endroit de la porte, comme dans la zone numéro 1 entourée sur la figure 3.12d. Ce n'est pas un problème pour notre solution de navigation *indoor* : il est intéressant de connaître le chemin à emprunter entre deux pièces. En revanche, certaines pièces sont complètement fermées, comme la zone numéro 2.

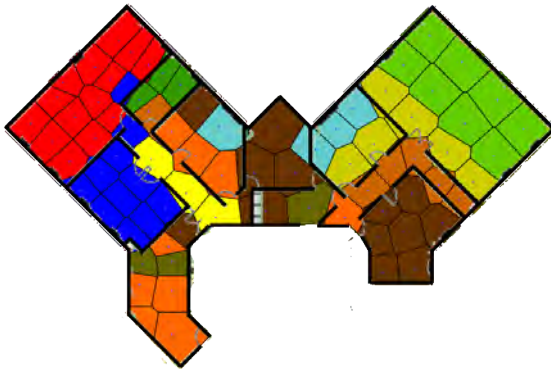
3.3 Identification des chemins

La navigation *indoor* nécessite de connaître la position de l'utilisateur, d'avoir le plan du bâtiment, mais aussi de connaître les chemins reliant tous les endroits du bâtiment. Dans le cas où certaines pièces sont isolées, il est donc nécessaire de connaître la position de la porte qui permet d'y entrer. De plus, il est nécessaire d'identifier le chemin pour atteindre toute destination à partir de la position de l'utilisateur. Cette section traite de ces deux points en particulier.

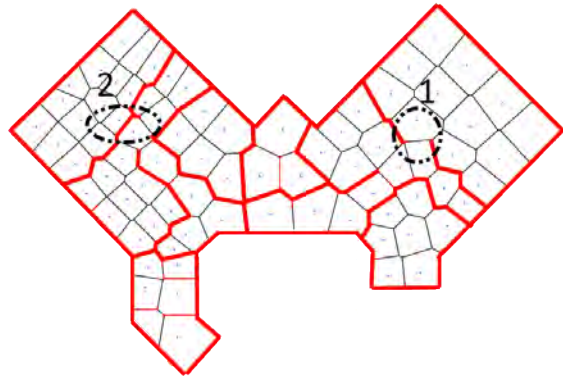


(a) Carte colorée obtenue en utilisant un partitionnement k-means avec $k = 12$.

(b) Carte colorée obtenue en utilisant un partitionnement k-means avec $k = 13$.



(c) Carte colorée obtenue en utilisant le partitionnement hiérarchique.



(d) Plan obtenu par InPreSS.

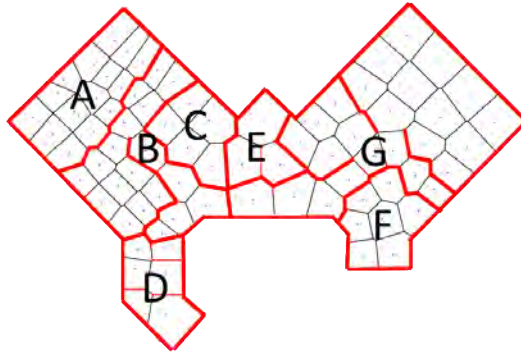
FIGURE 3.12 Comparaison des plans obtenus par quatre méthodes de classification dont InPreSS.

3.3.1 Détection des portes

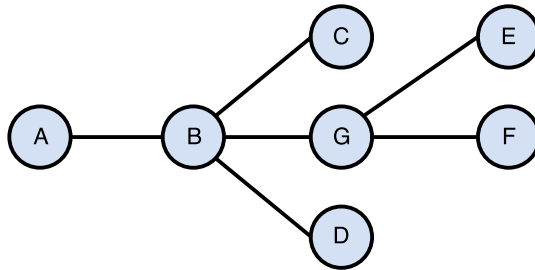
La première étape pour pouvoir proposer un chemin dans le bâtiment à l'utilisateur est de connaître les connexions entre les pièces fermées obtenues dans la section précédente : les portes.

Le plan que nous venons d'obtenir est, dans un premier temps, transformé en un graphe non-orienté entre les pièces. Chaque pièce fermée est représentée par un nœud. Ensuite, lorsqu'un utilisateur se déplace d'une pièce à l'autre, une arête est créée entre les

deux sommets qui les représentent. On obtient alors le graphe présenté en figure 3.13b.



(a) Carte engendrée par la section précédente.



(b) Graphe connecté obtenu à partir de la carte.

FIGURE 3.13 Carte et graphe obtenus par InPreSS.

Ce graphe de niveau macroscopique permet de retrouver tous les chemins possibles entre les pièces. En revanche, s'il existe plusieurs portes reliant deux pièces, InPreSS n'en détecte qu'une. Ce n'est pas un problème pour retrouver le chemin (car une porte existera toujours), mais cela peut rallonger certains trajets.

À un niveau plus resserré, chaque pièce est elle-même représentée par un graphe où chaque cellule de Voronoi est un sommet relié par une arête à chaque cellule adjacente. La figure 3.14b représente notamment le graphe non-orienté obtenu pour la pièce B.

Il reste à déterminer où se trouvent les liens entre les pièces, obtenus à l'échelle macroscopique, sur l'échelle microscopique. Pour cela, nous avons choisi de positionner la porte au niveau de la frontière entre les deux cellules de Voronoi voisines de pièces différentes dont la dissimilarité est la plus faible. Nous obtenons alors la carte présentée sur la figure 3.15.

Sur les graphes microscopiques de chaque pièce, il suffit alors de rajouter un lien entre

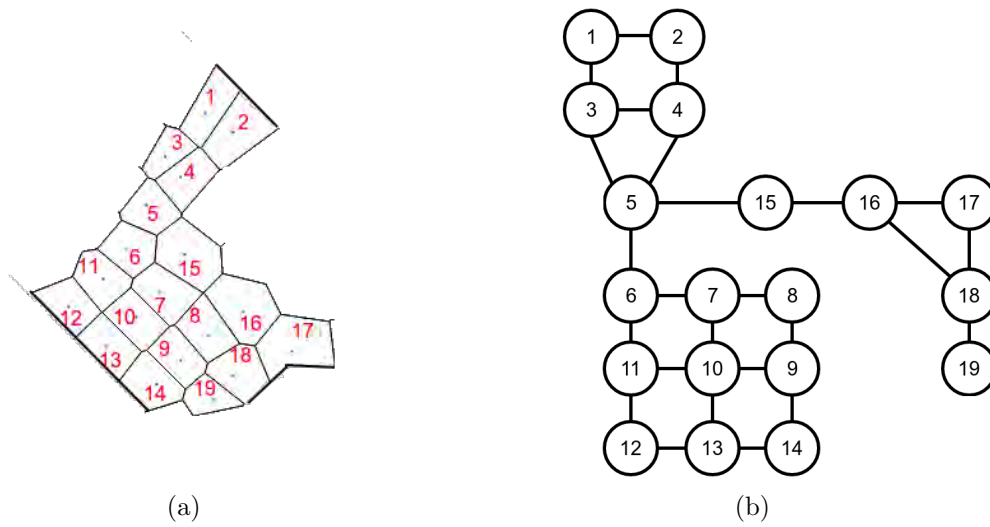


FIGURE 3.14 Graphe de la pièce B.

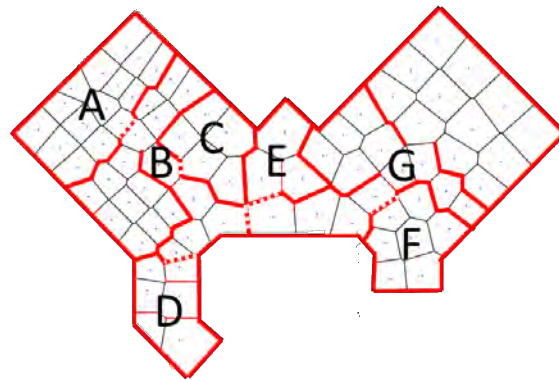


FIGURE 3.15 Carte finale obtenue par InPreSS. Les pointillés rouges représentent la localisation des “portes” estimée par l’algorithme.

la cellule de Voronoi qui contient la porte et le nœud représentant la pièce que la porte permet d’atteindre. La figure 3.16 représente le graphe microscopique de la pièce B et ses liens vers les autres pièces.

3.3.2 Du graphe au chemin

Lorsqu’un utilisateur veut rejoindre sa destination, il suffit de déterminer la cellule de Voronoi dont il dépend et celle de sa destination. Dès lors, le graphe macroscopique donne le chemin pour rejoindre la pièce de destination à partir de la pièce source. À

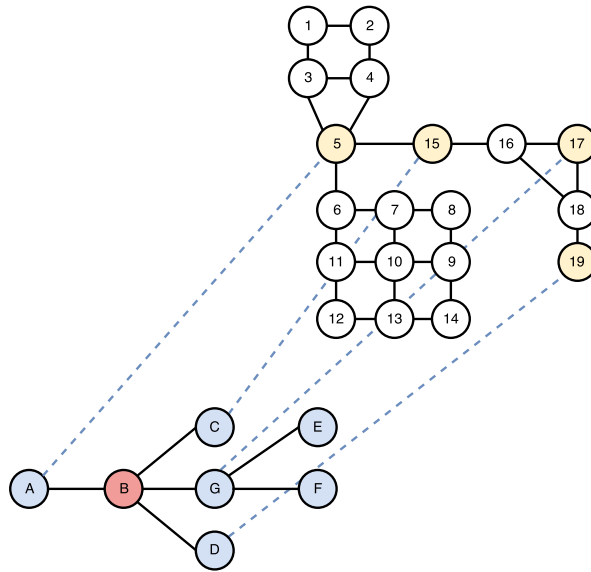


FIGURE 3.16 Graphe à deux échelles. L'échelle macroscopique permet d'établir les liens entre les pièces. L'échelle microscopique permet d'identifier les chemins dans chaque pièce.

l'intérieur de chaque pièce, le trajet est défini par le graphe microscopique, en utilisant l'algorithme de Dijkstra pour trouver le plus court chemin [116].

Le problème principal de ce modèle où chaque arête possède le même poids est que l'on risque de prendre un chemin plus long que nécessaire : toutes les cellules de Voronoi n'ont pas nécessairement la même taille. Un autre problème réside dans la particularité de nos déplacements dans un bâtiment : on ne recherche pas nécessairement le chemin le plus court, mais généralement celui qui nous fait traverser le moins de pièces. Nous avons donc décidé de découper le problème en deux. Dans un premier temps, nous recherchons le plus court chemin en termes d'arêtes parcourues au niveau macroscopique, afin de traverser le moins de pièces possibles. Dans un deuxième temps, le déplacement dans une même pièce est définie en trouvant le trajet le plus court. Pour cela, comme la position des germes des cellules de Voronoi sont connues, nous ajoutons un poids à l'arête reliant deux cellules équivalant à la distance séparant les deux germes.

3.4 Cas réaliste : le bâtiment inconnu

Dans cette section, nous évaluons InPreSS dans une utilisation concrète de l'application et sur un nouveau bâtiment, présenté sur la figure 3.17. Nous évaluons sa capacité à

construire le plan du bâtiment après le passage d'un utilisateur, puis d'un second. Enfin, nous évaluons la consommation énergétique de l'application.

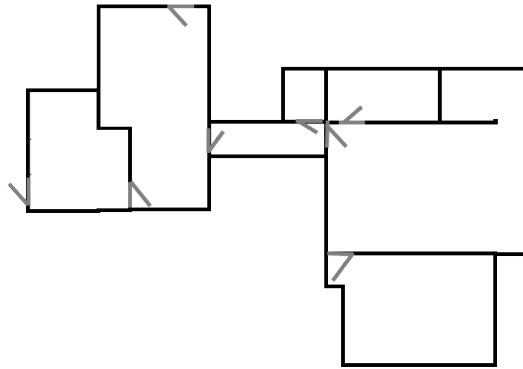


FIGURE 3.17 Plan du bâtiment-test.

Pour pouvoir proposer une navigation dans le bâtiment, une seule information est nécessaire : le contour extérieur du bâtiment, que l'on peut retrouver facilement grâce à des analyses satellites ou à travers une application comme Google Maps [25]. Lorsque le premier utilisateur entre dans le bâtiment, nous ne pouvons pas lui proposer de plan ni de navigation. En revanche, nous collectons ses données pour reconstruire un plan de navigation dans le bâtiment. Chaque point de collecte représente un germe dans le partitionnement de Voronoi. Pour InPreSS, un germe est collecté toutes les deux secondes. Cela permet d'obtenir environ un point tous les deux mètres lors du trajet de l'utilisateur. Il s'agit aussi de l'ordre de grandeur de la fréquence de rafraîchissement de la plupart des solutions de localisation *indoor* [18, 62].

Ce plan pourra être proposé aux prochains utilisateurs souhaitant naviguer dans le bâtiment. Ces mêmes utilisateurs enrichissent par leurs trajets le plan du bâtiment. Les cellules de Voronoi seront recalculées en fonction des nouveaux points de mesure (germes) des nouveaux utilisateurs. Il est par ailleurs possible que ces nouveaux points soient trop proches des anciens pour créer deux cellules différentes. Dans ce cas, on peut envisager de remplacer le germe le plus ancien par le nouveau, pris au même endroit.

3.4.1 Après le trajet d'un utilisateur

Lors de notre expérimentation, le premier utilisateur a suivi la trajectoire présentée en rouge sur la figure 3.18. Avec seulement vingt quatre points de mesure, cinq pièces sont correctement identifiées sur les huit du bâtiment. Il s'agit en réalité des cinq pièces visitées par l'utilisateur. Certaines pièces restent cachées de l'application. Ce sera le cas tant qu'aucun utilisateur n'y sera rentré. En pointillés, on peut également voir que les portes ont été correctement trouvées, même si elles ne sont pas précisément indiquées, par manque de points de mesure.

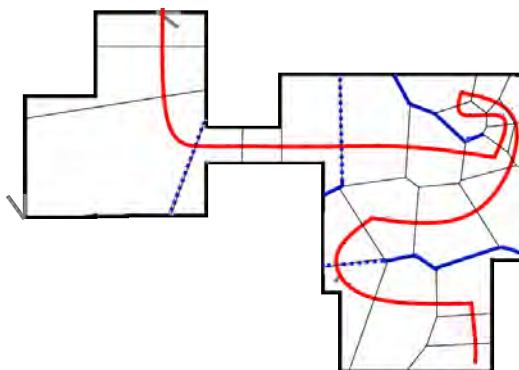


FIGURE 3.18 Trajet du premier utilisateur (en rouge) et plan obtenu (en bleu).

3.4.2 En regroupant les données de deux utilisateurs

Pour compléter ce plan, un deuxième utilisateur a suivi, quant à lui, une autre trajectoire, qui complète celle du premier utilisateur. Elle est présentée en rouge sur la figure 3.19. Ce dernier n'utilise pas le même smartphone que le premier utilisateur - un Samsung Galaxy S5 contre un LG Google Nexus 5 -, et il effectue son trajet à un autre moment de la journée : il est donc impossible de fusionner les deux bases de données. La luminosité dans le bâtiment a pu changer par exemple. Ainsi, InPreSS calcule un deuxième plan du bâtiment à l'aide uniquement des données du deuxième utilisateur. On obtient le plan en orange de la figure 3.19. Grâce à ce deuxième utilisateur, six pièces sur huit ont été identifiées par InPreSS.

Ensuite, InPreSS fusionne les deux plans calculés par superposition : on applique le diagramme de Voronoi sur l'ensemble des points de mesure localisés des deux utilisateurs. Pour chaque couple de cellules de Voronoi (A et B) adjacentes issues des deux plans

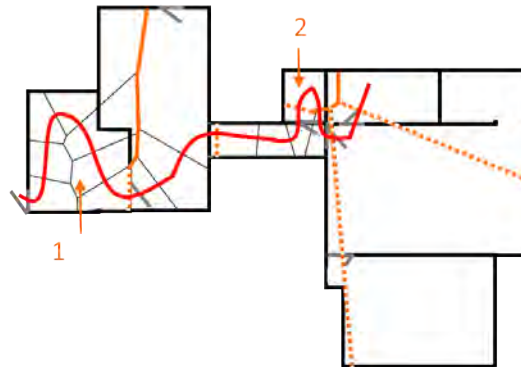


FIGURE 3.19 Trajet du second utilisateur (en rouge) et plan obtenu (en orange).

précédemment calculés, trois cas se présentent.

- Les deux cellules sont toujours adjacentes dans le nouveau découpage du plan : la dissimilarité entre les deux cellules est conservée (que ce soit un mur, une porte ou une ouverture).
- Les deux cellules ne sont plus adjacentes dans le nouveau découpage du plan et il n'existait pas de mur ou de porte : les nouveaux liens sont ouverts.
- Les deux cellules ne sont plus adjacentes dans le nouveau découpage du plan, et il existait un mur entre les deux cellules A et B. Ce mur est positionné entre les deux nouvelles cellules placées entre A et B les plus éloignées.

La figure 3.20 présente le plan du bâtiment obtenu (en rouge). Avec deux utilisateurs, sept pièces sur huit ont été correctement identifiées.

Pour obtenir un plan évolutif, c'est-à-dire qui s'adapte aux changements du bâtiment, il suffit donc de rajouter une durée de vie aux données remontées. Cette durée de vie ne peut être fixée dans l'absolu et doit dépendre de la taille du bâtiment et du nombre d'utilisateurs qui le visitent.

Pour notre dernière expérience, Bob arrive finalement dans notre bâtiment test, à l'entrée (voir figure 3.21. Il veut rejoindre le point en forme d'étoile. InPreSS lui propose au niveau macroscopique le trajet : B-C-E-F. Sur son application, il peut voir le plan du bâtiment et le trajet de cellule en cellule comme le montre la figure 3.21.

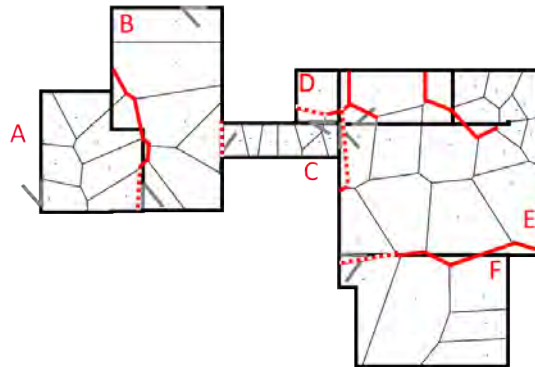


FIGURE 3.20 Plan du bâtiment obtenu à l'aide de deux utilisateurs.

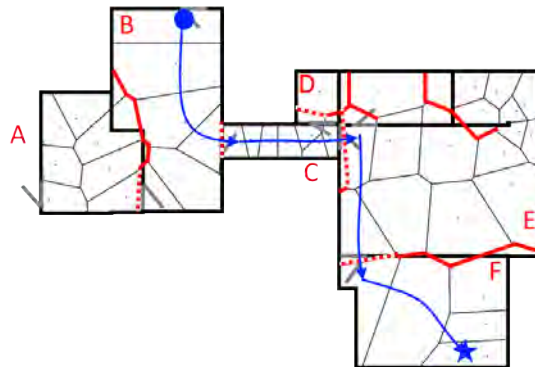


FIGURE 3.21 Trajet proposé (en bleu) par InPreSS pour rejoindre sa destination (étoile) à partir de sa position (cercle).

3.4.3 Consommation énergétique

Le plan calculé à partir des données du smartphone d'un utilisateur ne lui sert pas directement : il est utile pour les prochaines personnes qui entreront dans le bâtiment. La consommation de batterie sur le smartphone de l'utilisateur doit donc être limitée. Or, la consommation énergétique d'InPreSS est directement liée à celles des capteurs d'ambiance utilisés. La figure 3.22 montre le niveau de batterie d'un smartphone LG Google Nexus 5 en fonction du temps en utilisant InPreSS. La batterie de l'utilisateur n'a baissé que de 1,7% en deux heures.

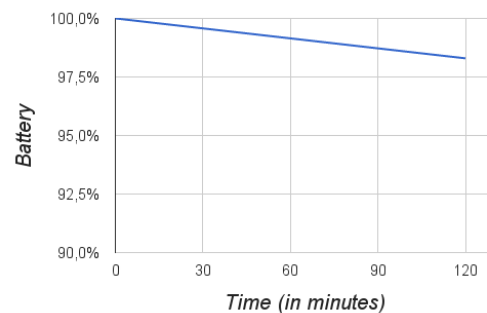


FIGURE 3.22 Consommation de batterie d’InPreSS au cours du temps.

3.4.4 Conclusion

InPreSS permet de reconstruire le plan d’un bâtiment grâce à l’aide des capteurs de smartphones des utilisateurs qui se déplacent dans toutes les pièces. En cumulant leurs trajets, InPreSS peut ainsi définir l’ensemble des chemins possibles entre tous les endroits du bâtiment.

Dès le passage du premier utilisateur, les données que son smartphone a collectées permettent d’élaborer un plan. Le suivant peut ainsi naviguer dans le bâtiment dans les pièces déjà empruntées par l’utilisateur précédent. Les passages réguliers de chaque utilisateur renforcent alors la quantité de chemins et de destinations possibles.

Dès lors, InPreSS propose un chemin entre les salles du bâtiment jusqu’à la destination choisie par l’utilisateur. Finalement, la collecte des informations des capteurs a peu d’influence sur la batterie de l’utilisateur.

EXTENSION DE LA DÉTECTION DE LOCOMOTION AUX MODES DE TRANSPORT

Notre étude sur la localisation *indoor* nous a conduit à l'analyse poussée des capteurs inclus dans nos smartphones. Notamment, nous avons pu observer que l'étude des signaux des capteurs de mouvement peut permettre de détecter les locomotions de l'homme. Pourquoi se limiter alors aux déplacements humains ? Nous tentons dans ce chapitre d'étendre la détection de locomotions humaines à l'ensemble de nos moyens de déplacement : vélo, voiture, train, etc. Nous montrons que les seuls capteurs de mouvement ne permettent pas d'obtenir une précision suffisante dans la détection de tous les moyens de transport. Or, comme nous avons pu le voir dans le chapitre précédent, les autres capteurs du smartphone peuvent aussi être utilisés pour l'analyse de l'environnement.

Dans ce chapitre, nous faisons une analyse des solutions de détection du mode de transport présentes dans la littérature. Nous proposons alors notre solution de détection de moyens de transport fondée sur un système de classification de type Random Forest [117]. Enfin nous envisageons une application concrète de ce type d'algorithmes à la gestion de places de parking en ville.

4.1 La détection du mode de transport

La détection du mode de transport n'a longtemps été qu'un sous-domaine de la recherche dans la détection d'activités humaines [118, 119]. Ces solutions consistent à déterminer l'occupation de l'utilisateur, notamment dans le suivi de ses activités sportives (course à pied, gymnastique, etc.) ou quotidienne (brossage de dents, cuisine,

etc.). Conformément à ces solutions, les premières méthodes de détection de mode de transport ont proposé de créer une carte embarquée par l'utilisateur formée de plusieurs capteurs [119, 120]. Rapidement, cependant, la place prépondérante des smartphones a remis celui-ci au centre des recherches sur la détection du mode de transport. Ces dernières années, des applications sont donc apparues pour fonctionner directement sur smartphone, sans équipement externe.

4.1.1 Les systèmes GPS

Les premières solutions sur smartphones envisagées reposent sur le GPS. La vitesse de déplacement de l'utilisateur qu'il est possible de calculer à partir des coordonnées GPS peut suffire pour distinguer un véhicule motorisé en vitesse de pointe, d'un vélo ou d'un piéton. Cependant, en ville, la vitesse d'une voiture est très irrégulière et dépendante de l'environnement et du trafic. Zheng et al. [121, 122] proposent d'utiliser d'autres informations, telles que les changements de direction ou les accélérations (calculées à partir de la vitesse). À partir de toutes ces informations et de la connaissance du réseau routier, les auteurs parviennent à une précision de 75,6% dans la classification des moyens de transports de l'utilisateur suivants : piéton, bus, voiture, vélo. Les travaux ultérieurs ont été consacrés à l'amélioration de cette précision, notamment avec l'ajout d'autres informations.

[123, 124] proposent d'utiliser les données extérieures, tels que les données géographiques via le système d'information géographique (SIG), les positions des bus, les lignes ferroviaires, etc. Ils parviennent ainsi à améliorer la précision de détection de 17%. Cependant, ces informations sont difficilement accessibles pour une application à grande échelle. C'est pourquoi Reddy et al. [125] ou Tao Feng et al. [126] ont proposé de combiner les données GPS aux accéléromètres embarqués sur les smartphones, permettant d'atteindre une précision de plus de 90% pour les modes de transport motorisés, stationnaires, pédestres et cyclistes. Ils utilisent notamment une méthode de classification de type "arbre de décision".

Toutes ces solutions possèdent un inconvénient majeur : la consommation excessive de batterie. En effet, le GPS est particulièrement coûteux en termes d'énergie et ne peut pas être utilisé en continu indéfiniment. De plus, le temps d'obtention du signal GPS est très variable et peut être un frein dans le temps d'obtention de la classification. Pire, le signal GPS peut être perdu (notamment en intérieur, bien sûr, comme dans les parkings souterrains).

4.1.2 Les signaux terrestres

Afin d'éviter l'utilisation du signal GPS, Muthukrishnan et al. [127] et le système LOCADIO [128] ont proposé d'utiliser le signal Wi-Fi. Cependant celui-ci est très instable et ces propositions se limitent à détecter la mobilité de l'utilisateur suivant les changements dans le signal Wi-Fi reçu. Ils ne distinguent donc pas spécifiquement les modes de transport. Sohn et al. [129] utilisent le signal GSM, plus stable et surtout présent partout. Pourtant, le taux de détection n'est toujours pas suffisant. Mun et al. [130] proposent ainsi de combiner les deux infrastructures : Wi-Fi et GSM. Ils parviennent à obtenir un taux de détection de l'ordre de 85% pour trois types de transport : voiture, piédestre et stationnaire.

Les signaux terrestres sont ainsi une bonne solution pour parvenir à identifier simplement si l'utilisateur est piéton ou motorisé. Cela limite par conséquent les applications, dans la mesure où elles ne permettent pas de différencier correctement les différents modes de transport.

4.1.3 Les accéléromètres

Wang et al. [131] proposent d'utiliser les accéléromètres présents dans les smartphones pour détecter les modes de transport suivants : stationnaire, piédestre, bus, train, métro, tram. Pourtant, le taux de détection n'est que de 60% en raison de l'incertitude des valeurs d'accélération. Peaks [132] est le premier système qui permette d'atteindre un taux de précision plus satisfaisant, de l'ordre de 85%. Pour cela, les auteurs proposent un algorithme de classification en trois temps. Tout d'abord, ils extraient du signal d'accélération une dizaine de paramètres : fréquence, moyenne, intensité, maxima, etc. Dans cette sélection réside leur principale différenciation, elle permet d'obtenir une information plus complète sur le signal qu'une simple valeur ponctuelle. Ils séparent ensuite les moyens de transports en trois types : motorisés, piédestres et stationnaires grâce aux informations simples de l'accélération. Enfin, le mode de transport précis (voiture, train, bus) est identifié grâce à une méthode de classification.

4.1.4 Objectifs

Les systèmes de détection de mode de transports fondés sur les réseaux satellitaires ou terrestres combinent imprécision et forte consommation énergétique. Les solutions utilisant les accéléromètres du smartphone sont prometteuses puisqu'elles parviennent à atteindre un taux de détection de l'ordre de plus de 80% tout en ayant un impact limité sur la batterie de l'utilisateur. Pourtant, certaines applications nécessitent un taux de

détection plus important encore, et ce, quel que soit le mode de transport de l'utilisateur.

Comme nous avons pu le voir dans les chapitres précédents, les smartphones récents contiennent de plus en plus de capteurs. Notamment, la plupart de ceux qui devaient être embarqués séparément pour la détection du mode de transport [133] sont aujourd'hui présents dans les smartphones. Nous proposons donc d'utiliser l'ensemble des capteurs des smartphones pour détecter le mode de transport de l'utilisateur, sans se limiter aux seuls accéléromètres.

4.2 Détection du mode de transport avec les capteurs du smartphone

Dans cette section, nous proposons notre module de détection du mode de transport sur smartphone. Celui-ci nécessite le signal des principaux capteurs embarqués, puis indique le mode de transport de l'utilisateur, parmi : **piéton**, **vélo**, **bus**, **voiture**, **métro**, **moto**, **train**, **tram** et **avion**. La précision de l'algorithme, supérieure à 95%, est obtenue grâce à l'extraction des principales statistiques des signaux des capteurs et à un algorithme de classification adaptée, de type Random Forest [117].

4.2.1 Les signaux captés par le smartphone

Le fait que l'accélération d'un utilisateur soit un indice sur son mode de transport paraît intuitif : un piéton ne peut pas accélérer aussi vite qu'une voiture. La figure 4.1 propose de confirmer notre intuition. Nous avons installé sur les smartphones de douze utilisateurs une application de collecte des signaux des capteurs d'ambiance. Pour différents moyens de transport, nous avons alors collecté les valeurs des capteurs. Sur la figure 4.1, nous avons représenté la répartition de l'accélération frontale (moyenne sur 2s) dans les différents modes de transport considérés, sous forme de boîte à moustaches (boxplot). La plus haute valeur médiane est obtenue pour le métro, ce qui peut s'expliquer par les fortes accélérations successives subies par l'utilisateur d'un métro après chaque arrêt. En revanche, on peut s'apercevoir que la différence est beaucoup moins évidente entre la voiture et le reste des transports. Certes, l'accélération peut être importante en voiture, au démarrage, mais elle peut aussi être beaucoup moins forte durant le trajet ou dans un embouteillage. Ceci explique la répartition très large des valeurs de l'accélération en voiture. L'accélération du piéton est elle, en revanche, confinée à de faibles valeurs, en raison de ses limites physiologiques.

Ce même type d'intuition a été la clé de notre étude. Par exemple, le niveau sonore

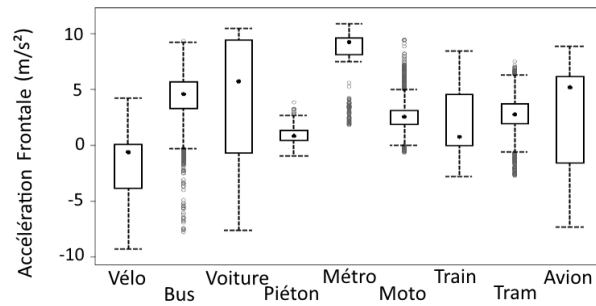


FIGURE 4.1 Boxplot de l'accélération frontale moyenne pour les 9 modes de transport considérés.

est bien plus élevé à moto qu'en bus ou en voiture. De la même façon, la luminosité est plus grande en extérieur qu'en intérieur, ce qui se répercute sur le capteur de lumière du smartphone lorsqu'il est dans un bus ou dans un métro.

La figure 4.2 présente la répartition du niveau sonore dans les différents modes de transport considérés, sous forme de boîte à moustaches (ou boxplot) et sur plusieurs heures d'expérimentation. Toutes les mesures ont été obtenues en utilisant un LG Google Nexus 5. L'intuition que nous avons eue est en partie confirmée : le niveau sonore médian est bien plus élevé à moto qu'en bus ou à pied. De plus, l'ensemble des valeurs est concentré autour de cette valeur médiane pour la moto. Cela permet d'obtenir une corrélation relativement élevée entre la valeur du niveau sonore capté par le smartphone et le transport en moto. Pourtant, en avion (il s'agit plus précisément de planeurs de type ULM) ou en voiture, le niveau sonore peut également être élevé. Par ailleurs, certains moyens de transport ne sont pas aussi bien identifiés : on peut voir qu'un piéton peut rencontrer tout type de niveau sonore, suivant sa proximité avec la route notamment.

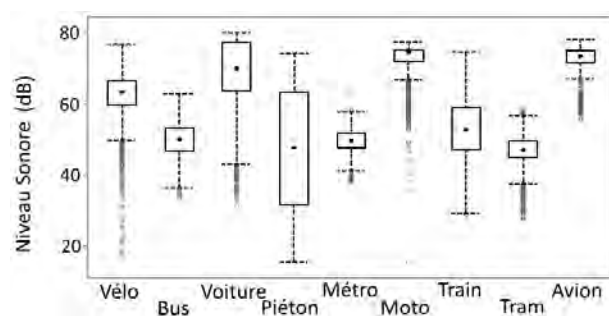


FIGURE 4.2 Boxplot du niveau sonore pour les 9 modes de transport considérés.

L'analyse de ces deux signaux permet d'obtenir une information non-négligeable sur le mode de transport de l'utilisateur. Une valeur ponctuelle d'un des capteurs n'est pourtant

pas suffisante pour le déterminer avec précision. On peut également voir la limite des solutions fondées uniquement sur les accéléromètres du smartphone : l'accélération seule ne permet pas d'identifier à coup sûr le mode de transport. La première idée évidente est donc de combiner les deux capteurs pour obtenir plus d'information. La figure 4.3 propose un graphique combiné des valeurs d'accélération et de niveau sonore obtenus dans les différents modes de transport. Chaque point est un couple de valeurs et chaque couleur représente un mode de transport. Sur ce graphique, à l'aide de seulement deux seuils, on peut aisément distinguer les valeurs obtenues en train, en avion ou en voiture des autres modes de transport. Cela confirme l'idée que la combinaison de capteurs améliore grandement la détection du mode de transport.

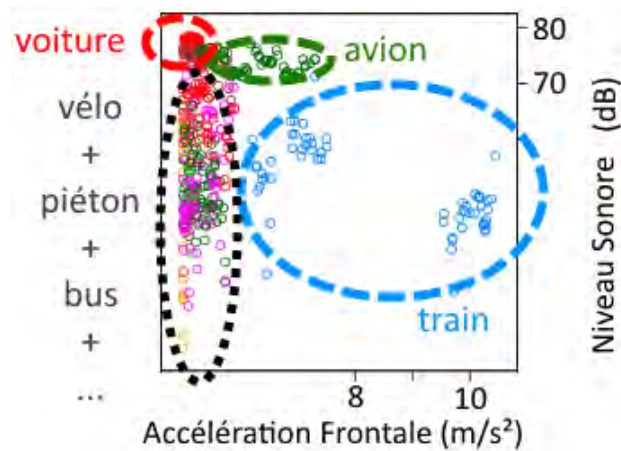


FIGURE 4.3 Graphique des valeurs de niveau sonore et d'accélération appairés.

Pourtant, on peut voir que deux capteurs ne suffisent toujours pas pour distinguer tous les modes de transport envisagés. De plus, comme le propose [132], il est possible d'extraire plus d'information d'un capteur que sa simple valeur moyenne. Ainsi, notre méthode de détection de mode de transport propose de combiner les signaux de tous les capteurs communs aux smartphones récents :

- les accéléromètres, dans les trois directions ;
- les gyroscopes, dans les trois directions ;
- l'orientation, dans les trois directions ;
- le champ magnétique, dans les trois directions ;
- le capteur de luminosité ;
- le baromètre ;
- le microphone, pour le niveau sonore.

Nous obtenons donc un total de 15 signaux. Pour chaque signal à trois axes, le signal

est transformé dans le référentiel de l'utilisateur, comme dans les chapitres précédents.

4.2.2 Un système de classification approprié : *Random Forest*

Peaks [132] a mis en évidence certaines caractéristiques du signal d'accélération utiles dans la détection du mode de transport : les moyennes, les extrema, etc. Nous nous sommes donc inspirés de cette liste pour établir une liste non-exhaustive des paramètres qui semblent pertinents.

Collection des variables explicatives

Le tableau 4.1 montre les 64 statistiques retenues dans chaque domaine temporel, fréquentiel et par ondelettes et pour chaque capteur. Cela conduit à 2880 variables explicatives.

Domaines	Temporel, fréquentiel, ondelettes
Statistiques générales	Moyennes, écart type, variance, médiane, valeur efficace, quartiles
5 pics et 5 vallées	Valeur, magnitude, largeur à mi-hauteur, 2 inter-distances
Forme	Kurtosis, asymétrie, entropie, surface

TABLE 4.1 Statistiques extraites de chaque capteur pour créer les variables du modèle.

Nous avons choisi d'utiliser les trois domaines principaux. Le domaine temporel, le plus classique, permet d'obtenir les principales caractéristiques connues des signaux, telles que la moyenne, les extrema, etc. Le domaine fréquentiel permet de calculer la fréquence (ou la période) du signal, comme proposé par [132]. Mais plutôt que de se limiter à cette unique valeur, nous observons l'ensemble du spectre fréquentiel, représentatif d'un signal. Le domaine en ondelettes est utilisé pour extraire notamment les informations des phénomènes locaux ou des discontinuités du signal.

Dans ces trois domaines, nous avons séparé les 64 statistiques en trois types.

- **Statistiques communes** : il s'agit des statistiques les plus connues, incluant notamment l'ensemble des moyennes (géométriques, harmoniques, arithmétiques) ainsi que les quartiles.
- **Pics et Vallées** : nous extrayons du signal les cinq pics et cinq vallées principaux. Pour chacun d'entre eux, nous déterminons l'abscisse du pic (la fréquence dans le domaine fréquentiel), sa magnitude (l'extremum dans le domaine temporel), sa largeur à mi-hauteur, et la distance au prochain pic ou à la prochaine vallée dans les deux sens.

- **Forme** : nous avons également choisi de calculer le kurtosis, qui mesure l'aplatissement du signal, l'asymétrie, l'entropie et la surface. Ces paramètres de forme permettent d'identifier la forme du signal et non une valeur locale ou moyenne.

Systeme de classification

Plusieurs solutions de classification ont été proposées dans la littérature. Le plus connu d'entre eux est appelé *Partial Least Squares* (PLS) [134], qui est lui-même une évolution de l'analyse en composantes principales (ACP). Il essaie de définir un modèle linéaire [135] entre les variables explicatives et les variables cibles - qui sont, dans notre cas, les moyens de transport. Ce modèle, calculé de façon itérative, permet d'extraire les variables latentes (non-observables) qui décrivent au mieux les variables cibles. PLS s'est montré par conséquent particulièrement efficace dans les cas où l'espace des variables explicatives est bien plus grand que celui des variables cibles, ce qui est notre cas (2880 pour 9). En effet, il est possible de choisir le nombre de composants créés pour décrire l'espace cible. Pourtant, comme PLS recherche des combinaisons linéaires entre les variables explicatives, il faut qu'il puisse y en avoir [136] pour obtenir les variables cibles. Or la diversité des variables explicatives que l'on a pu obtenir rend l'objectif ambitieux. Cela résulterait probablement en la création d'un trop grand nombre de composants rendant inefficace l'analyse.

Nous nous sommes donc plutôt tournés vers une approche de type arborescente. Les arbres de décision sont des structures hiérarchiques à nœuds. À chaque nœud de l'arbre, l'algorithme sélectionne une variable explicative et un seuil. Elle classe ensuite les expériences en deux branches : celles dont la variable calculée est inférieure au seuil et l'autre. La variable et le seuil ne sont pas choisis au hasard : ce sont ceux qui partitionnent le mieux les expériences, c'est-à-dire que le désordre dans les deux branches est minimisé. Dans notre contexte, le désordre correspond au fait d'avoir le plus faible nombre de modes de transport différents dans chaque branche, de sorte que les observations d'un même mode de transport partent dans la même branche. Pour estimer ce désordre, nous avons choisi d'utiliser l'entropie :

$$D_i = - \sum_{m \in \forall \text{modes}} \frac{n_{m,i}}{N_i} \log\left(\frac{n_{m,i}}{N_i}\right)$$

où N_i est le nombre d'observations dans le nœud i et $n_{m,i}$ le nombre d'observations correspondant au mode de transport m dans le nœud i .

Dans les nœuds terminaux, il ne doit alors rester que les observations appartenant au même mode de transport. La figure 4.4 montre un exemple de branche finale d'arbre

obtenue.

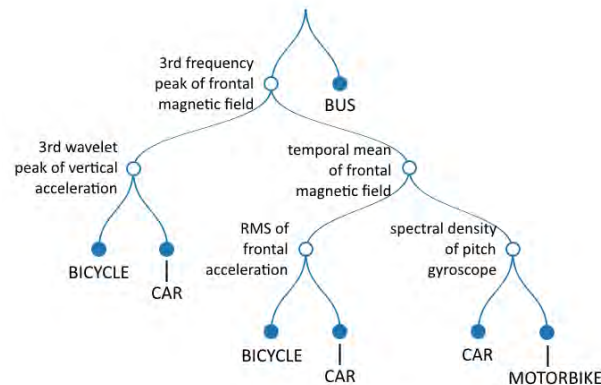


FIGURE 4.4 Une des branches finales d'un des arbres obtenus en utilisant l'algorithme de Random Forest

Bien qu'un arbre unique puisse fonctionner, il n'est pas adapté à un grand nombre de variables explicatives, dans la mesure où une seule d'entre elles est choisie à chaque nœud. Pour résoudre ce problème, nous avons choisi d'utiliser l'algorithme Random Forest [117]. À chaque nœud, il sélectionne la meilleure variable explicative parmi une sous-partie seulement de l'ensemble des variables. Cette sous-partie est tirée aléatoirement. Par ailleurs, plutôt que de se contenter d'un arbre, Random Forest construit une forêt. Chaque arbre de la forêt est décorréolé des autres par ce tirage aléatoire à chaque nœud. Un autre avantage de l'algorithme de Random Forest est qu'il est facilement adaptable en temps réel, comme l'a montré [137].

Construction des arbres

En utilisant cette approche, nous avons mené une campagne d'expérimentations pour assembler une base de données importante. Nous avons proposé à douze utilisateurs de participer à la collecte, en essayant d'être représentatif d'un échantillon de la population. Les participants sont :

- 8 hommes et 4 femmes ;
- de 21 à 58 ans ;
- de 1m60 à 1m90.

Nous avons installé sur leur smartphone une application Android de collecte de données de capteurs. Les smartphones utilisés sont :

- Sony Xperia E3 ;
- LG Google Nexus 5 ;

- LG Google Nexus 6 ;
- LG G2 ;
- Huawei Honor 4X ;
- Samsung Galaxy S5 ;
- Samsung Galaxy S3 mini.

Nous avons ensuite demandé aux utilisateurs de rentrer régulièrement, sur leur smartphone, leur moyen de transport. Pendant ce temps, automatiquement, l'application collecte à la fréquence maximale les signaux des 15 capteurs du smartphone. Plusieurs mois plus tard, nous avons récolté l'ensemble des données récoltées qui représentent près de 30h d'expérimentation, dans trois zones géographiques différentes (Toulouse, Caen, Aix-en-Provence).

Sur l'ensemble de ces données, les 2880 variables explicatives sont calculées toutes les 2 secondes. Cette période de deux secondes permet d'obtenir suffisamment de valeurs différentes sans qu'elles ne soient trop corrélées. Avant d'appliquer le modèle de classification, nous avons décidé d'enlever certaines variables pour limiter le temps de calcul et supprimer les éventuelles corrélations et qui pourraient fausser l'algorithme. Ainsi, nous avons calculé la corrélation croisée entre toutes les variables explicatives et rejeté les plus corrélées. Cela nous permet d'obtenir une matrice de prédiction complète. C'est cette matrice qui est l'entrée de l'algorithme de Random Forest. Nous pouvons alors lancer l'analyse pour obtenir 500 arbres de la forêt, ordre de grandeur recommandé par [117].

Ces arbres sont calculés hors-ligne. Dans l'algorithme en temps réel, il suffit de calculer les variables et de parcourir les arbres. Le mode de transport le plus choisi par la forêt est celui identifié par notre algorithme.

Nous avons par ailleurs vérifié et analysé l'importance des statistiques retenues dans notre algorithme de classification. Pour obtenir cette importance, nous utilisons la formulation de [117] qui mesure la diminution du désordre $\delta(D(n))$ apportée par une variable explicative X_k dans chaque nœud n qui l'utilise et sommée sur l'ensemble des nœuds. Soit $I(X_k)$ l'importance de la variable X_k : $I(X_k) = \frac{1}{N} \sum_{n=1}^{N_k} \delta(D(n))$ où N est le nombre total de nœuds et N_k le nombre de nœuds qui utilisent la variable explicative X_k .

La figure 4.5 présente le taux d'importance des variables sommé par domaine. On peut voir que les trois domaines sont importants même si les domaines temporels et fréquentiels sont prépondérants. C'est probablement dû à la sélection des statistiques qui peut être plus pertinente dans les domaines que l'on connaît le mieux.

La figure 4.6 montre le taux d'importance des variables par type. Cela montre que

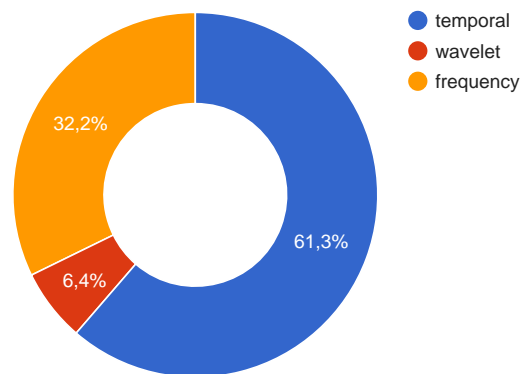


FIGURE 4.5 Importance des domaines choisis.

les trois types sont utiles dans la détermination du type de transport.

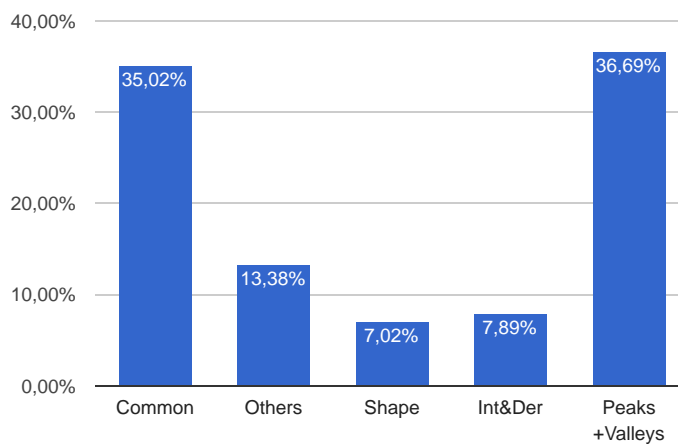


FIGURE 4.6 Importance des statistiques par type.

4.2.3 Performances

Pour valider les performances de ce modèle, nous avons demandé aux douze utilisateurs de reconduire l'expérience de collecte, pour obtenir 28 nouvelles heures d'expérimentation, cette fois pour tester le modèle. Au total 10282 décisions de mode de transport sont prises.

Dans un premier temps, nous avons choisi de vérifier notre choix d'utiliser l'algorithme de Random Forest par rapport à une approche PLS ou d'arbre de décision simple. Conformément à la théorie, le tableau 4.2 montre que Random Forest est de loin l'algorithme qui permet les meilleures performances, avec une précision de 98.65% largement supérieure aux autres méthodes. Pour exclure la probabilité de chance dans le choix du mode de transport, nous avons également calculé le critère kappa de Cohen défini par :

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

où p_o est la précision absolue et p_e représente la probabilité d'un accord aléatoire.

Méthode	Détection	Kappa
PLS	0,69	0,67
Single Tree	0,54	0,53
Random Forest	0,99	0,98

TABLE 4.2 Performances de Random Forest comparé aux autres méthodes de classification.

Dans notre cas, la précision absolue et le kappa sont très proches dans la mesure où le nombre d'expériences par mode de transport est plus ou moins équitablement réparti comme le montre le tableau 4.3.

Nous comparons ensuite la précision de notre algorithme dans la détection des neuf modes de transports indépendamment. Le tableau 4.3 présente la matrice de confusion obtenue pour les neuf modes de transport. La matrice montre notamment que la précision de la détection ne descend jamais en-dessous de 95% quel que soit le mode de transport.

Nous avons choisi de comparer notre solution aux principales solutions de la littérature : Zheng [121] pour les solutions GPS, Stenneth [123] pour les solutions GPS et informations extérieures, Reddy [125] pour les solutions GPS et accéléromètres, Wang [131] et Peaks [132] pour les solutions par accéléromètres uniquement. Le tableau 4.4 montre que notre solution est de loin la plus précise et la plus complète en termes de moyens de transport

Transport	Vélo	Bus	Voiture	Métro	Moto	Piéton	Train	Tram	Avion	Précision
Vélo	802	0	2	0	0	2	0	0	0	99,50%
Bus	0	837	0	0	0	0	0	47	0	97,51%
Voiture	3	8	1747	0	0	6	0	0	0	99,04%
Métro	0	1	0	583	0	0	0	26	0	95,57%
Moto	3	0	7	0	835	7	0	0	0	98,00%
Piéton	0	0	0	0	0	973	0	0	0	100,00%
Train	0	0	0	0	0	0	830	0	0	100,00%
Tram	0	7	0	0	0	0	0	1506	0	99,54%
Avion	0	7	0	0	0	0	0	6	1037	98,76%

TABLE 4.3 Matrice de confusion de notre solution de détection de mode de transport.

déTECTÉS.

Transport	Vélo	Bus	Voiture	Métro	Moto	Piéton	Train	Tram	Avion	Motorisé
Solution	99%	98%	99%	96%	98%	100%	100%	99%	99%	99%
Peaks	-	78%	-	65%	-	95%	68%	84%	-	74%
Wang	81%	58%	72%	53%	-	81%	-	-	-	54%
Reddy	87%	-	-	-	-	95%	-	-	-	95%
Zheng	66%	67%	86%	-	-	89%	-	-	-	76%
Stenneth	89%	88%	88%	-	-	98%	98%	-	-	91%

TABLE 4.4 Comparaison entre notre solution et les solutions concurrentes de la littérature.

4.3 SmartPark, une application concrète : les places de parking intelligentes

Cette section présente une application concrète de la détection du mode de transport : le stationnement dans les villes intelligentes.

Carole se rend en ville avec son véhicule préféré (la voiture, le vélo ou la moto) et cherche désespérément une place de parking libre. Elle allume son application sur son

smartphone qui lui propose une place libre, pas très loin d'elle et adaptée à son véhicule. Elle a gagné du temps et de l'argent. Cependant, cette application n'existe pas encore.

Pourtant, la recherche d'une place de parking en ville est un problème connu. Une étude du cabinet Parsons Brickerhoff [138] effectuée dans le quartier de Chinatown à New York City a montré que 54% des conducteurs passaient plus de 20min à rechercher une place de parking libre en ville. Ces conducteurs en recherche de stationnement représentent 30% du trafic en ville [139, 140]. Limiter cette recherche permettrait donc également de limiter l'empreinte carbone des villes.

Une fois garée, Carole peut vaquer à ses occupations. Lorsqu'elle choisit de libérer sa place de parking, l'application calcule automatiquement le temps passé sur le stationnement et, le cas échéant propose le règlement de la somme due. Cela permet à Carole de ne pas avoir à calculer le temps qu'elle va passer sur le stationnement et payer en conséquence le parcmètre. Dans le même temps, cela permet à la ville de récupérer une quantité non-négligeable de places impayées. Cincinnati, par exemple, a estimé à 12 millions de dollars le manque à gagner sur les places de parking non réglées en 2005 [141].

La voiture n'est pas le seul mode de transport concerné. De nombreuses villes aujourd'hui poussent, pour un gain de place et une diminution de la pollution, à l'utilisation de vélos et de motos. La ville de Portland en Oregon a proposé une carte des places de stationnement pour vélos [142]. Celles-ci sont rapidement prises d'assaut aux heures de pointe, ce qui empêche une démocratisation de l'utilisation du vélo. À Amsterdam par exemple, 73000 vélos mal garés ont dû être enlevés de la ville [143]. Une application de parking intelligent en ville doit donc aussi proposer une solution pour ces modes de transport alternatifs.

L'obstacle majeur d'une telle application est la détection des moments où l'utilisateur gare son véhicule et le moment où il libère la place... sans avoir réduit la batterie du smartphone de l'utilisateur à néant. Cette section propose d'analyser les solutions existantes, puis envisage une solution qui utilise le module de détection du mode de transport présenté précédemment.

4.3.1 Les solutions existantes

La première solution envisagée pour répondre à ce problème a été proposée par Google via Google Open Spot [144]. Fondée sur le volontariat, elle propose à l'utilisateur de rentrer lui-même dans l'application la place qu'il libère. Les études de Google ont

montré que le système est peu fiable à cause de la négligence de l'utilisateur et surtout non-envisageable dans le cas où la place est payante. Le GPS a également été proposé pour répondre au problème : l'application détecte le stationnement et la libération grâce à la localisation de l'utilisateur et à son mode de transport [123, 126]. Ces solutions trouvent leur limite dans la consommation excessive de batterie : le GPS doit être constamment allumé, même lorsque l'utilisateur est garé. Parksense [139] propose une solution qui repose uniquement sur le réseau Wi-Fi afin de réduire la consommation d'énergie. Malheureusement, nous avons pu voir précédemment que la précision du Wi-Fi est faible dans la détection du mode de transport, comme dans la localisation.

Faute d'une solution sur smartphone efficace, certaines villes ont copié le fonctionnement des parkings privés en installant des équipements spécifiques (capteur de poids, caméras, etc.) sur chaque place pour identifier les places libres. Cette solution a par exemple été choisie par San Francisco [145]. Le revers de la médaille de cette solution efficace est le coût d'installation et de maintenance. Une étude du département américain des transports [146] a estimé le coût de déploiement total de cette solution à San Francisco entre 7,2 et 23 millions de dollars pour 40000 places. Une solution sur smartphone est donc attendue pour limiter les investissements des villes connectées.

4.3.2 Architecture de SmartPark

Notre solution, appelée SmartPark, est le premier système qui permette de détecter la libération d'une place de parking, en se reposant uniquement sur le smartphone, et en ayant un impact négligeable sur la batterie. La figure 4.7 propose une bande dessinée illustrant le scénario de SmartPark du point de vue de l'utilisateur. Celui-ci installe SmartPark sur son smartphone. Lorsqu'il recherche une place de parking libre, il envoie une requête à l'application. Celle-ci détecte son mode de transport et lui envoie la liste des places disponibles autour de lui adaptées à son mode de transport. Il en choisit une et se gare. Il peut ensuite vaquer à ses occupations jusqu'à ce qu'il décide de libérer sa place. Automatiquement, le montant dû lui est prélevé en fonction du temps passé en stationnement.

La figure 4.8 présente l'architecture de SmartPark, qui suit quatre étapes majeures.

1. L'utilisateur recherche une place de parking et envoie une requête de stationnement libre : SmartPark utilise le module de détection de mode de transport et détermine les places libres dans le voisinage pour ce mode.
2. L'utilisateur se gare : SmartPark scanne les signaux Wi-Fi et cellulaires environ-

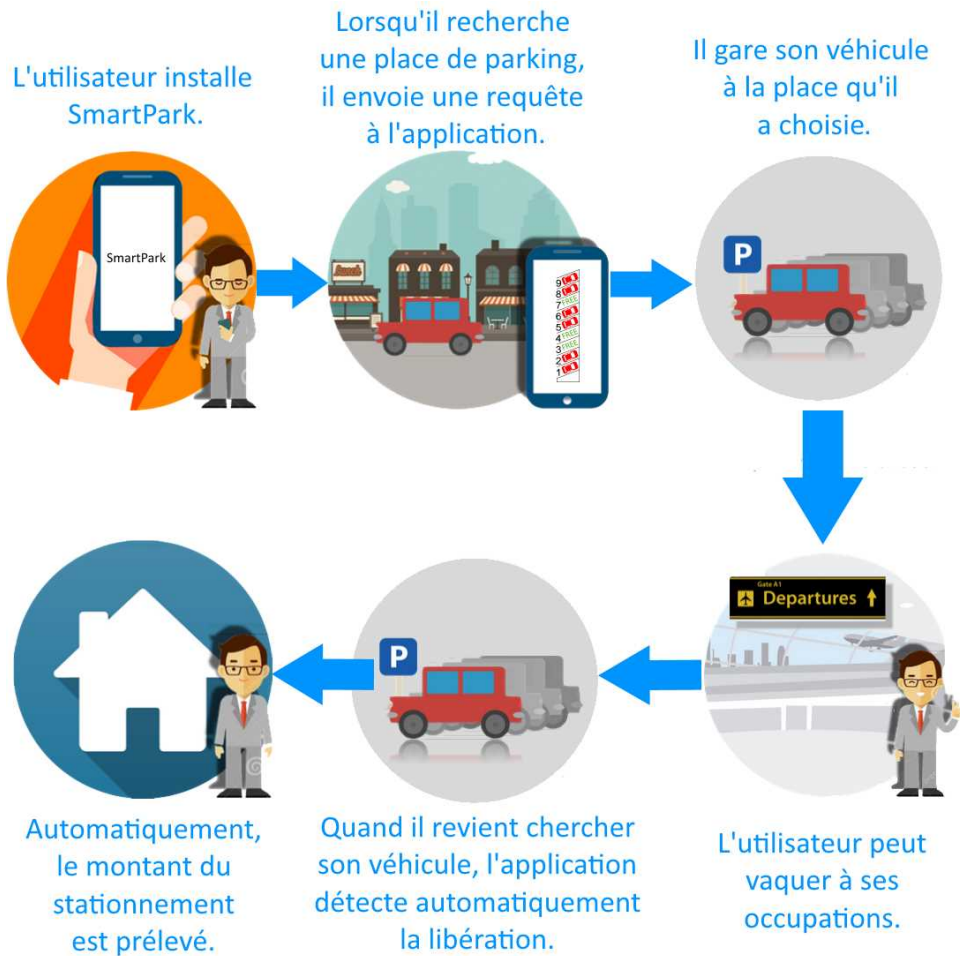


FIGURE 4.7 Le point de vue de l'utilisateur sur SmartPark.

nants afin de créer un profil de localisation. Celui-ci est sauvegardé pour une utilisation future.

3. Le véhicule est garé : l'utilisateur vaque à ses occupations. Il peut se promener, prendre le bus, le métro ou tout autre moyen de transport. SmartPark, dans le même temps, évalue toutes les 10s le mode de transport de l'utilisateur. Dès que l'utilisateur a repris un véhicule du même type que celui garé, une alerte est levée : l'utilisateur est peut-être en train de reprendre son véhicule. Pour vérifier qu'il s'agit bien du même véhicule - et non d'un véhicule du même type, récupéré plus loin, comme lors d'un covoiturage -, SmartPark scanne l'environnement et le compare au scan du lieu de stationnement.
4. L'utilisateur libère sa place de parking : SmartPark détecte cet événement car

l'utilisateur a repris un véhicule du même type et il l'a repris au même endroit d'après le scan.

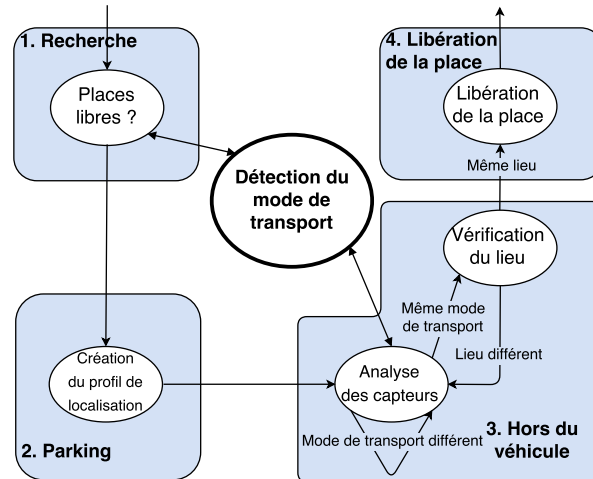


FIGURE 4.8 Architecture logique de SmartPark.

SmartPark repose ainsi sur deux fonctionnalités majeures :

- la détection du mode de transport, déjà présentée précédemment ;
- la vérification de la libération du véhicule.

4.3.3 Vérification de la libération du véhicule

Lorsqu'un utilisateur gare son véhicule, il entre le numéro de la place de parking qu'il a choisie dans son application. Automatiquement, SmartPark crée un profil de localisation en effectuant un scan Wi-Fi et cellulaire. Ce scan consiste à recenser les noms des réseaux à proximité et le niveau de signal reçu.

$$\mathbf{S}_1 = (S_1(1), S_1(2), \dots, S_1(a))$$

$$\mathbf{W}_{S_1} = (W_{s_1}(1), W_{s_1}(2), \dots, W_{s_1}(a))$$

$$\mathbf{C}_1 = (C_1(1), C_1(2), \dots, C_1(b))$$

$$\mathbf{W}_{C_1} = (W_{c_1}(1), W_{c_1}(2), \dots, W_{c_1}(b))$$

où \mathbf{S}_1 est la liste des SSID des a points d'accès à portée ; \mathbf{W}_{S_1} est la liste des niveaux de puissance de signaux reçus ; \mathbf{C}_1 est la liste des Cell-id des b stations captées ; \mathbf{W}_{C_1} les RSSI (*Received Signal Strength Indication*) des cellules voisines.

Dès lors, lorsque SmartPark détecte que l'utilisateur a repris le même mode de

transport que le véhicule garé, il effectue un nouveau scan, et compare le profil de localisation à celui précédemment collecté. Pour comparer, nous proposons un coefficient de similarité C :

$$C = \frac{1}{\sum_{i=1}^a W_{s_1}(i)} \sum_{i=1}^a W_{s_1}(i) \left(1 - \left| \frac{W_{s_1}(i) - W_{s_2}(i)}{W_{s_1}(i)} \right| \right) \quad (4.1)$$

$$+ \frac{1}{\sum_{i=1}^b W_{c_1}(i)} \sum_{i=1}^b W_{c_1}(i) \left(1 - \left| \frac{W_{c_1}(i) - W_{c_2}(i)}{W_{c_1}(i)} \right| \right)$$

où $W_{s_2}(i)$ et $W_{c_2}(i)$ représentent les niveaux de signaux du second scan correspondant aux points d'accès $S_1(i)$ et $C_1(i)$ - ramené à 0 s'il n'est pas trouvé.

Ce coefficient prend en compte trois éléments clés :

1. la proportion de points d'accès trouvés : plus le nombre de points d'accès retrouvés lors du second scan est grand, plus il est probable que les scans ont été effectués au même endroit ;
2. la valeur des niveaux de signal : plus la différence entre les deux mesures d'un même point d'accès est faible, plus le coefficient est grand.
3. favoriser les puissances les plus hautes : plus le point d'accès est proche, plus la puissance reçue est élevée, plus son poids dans le coefficient est important. L'idée est que si un point d'accès est proche lors du premier scan, il a d'autant plus de chances d'être présent lors du second scan, car il est proche du stationnement et subit donc moins la variabilité du signal Wi-Fi.

Pour vérifier les performances de ce scan, nous avons demandé à un volontaire de se garer puis de libérer sa place de parking à 29 reprises. Par ailleurs, nous lui avons demandé de réaliser de fausses libérations. Pour cela, le volontaire gare son véhicule à 27 reprises puis récupère un autre véhicule du même type un peu plus loin. Dans la figure 4.9, nous comparons le taux de détection (parmi les 29 événements de parking/libération) et de faux positifs (mauvaise détection de libération parmi les 27 faux-positifs) pour plusieurs coefficients de la littérature : ceux de Parksense [139], le traditionnel indice de Tanimoto [147] (pondéré ou non) et l'indice de Jaccard, le plus utilisé pour comparer deux scans Wi-Fi : $J = \frac{|\mathbf{S}_1 \cap \mathbf{S}_2|}{|\mathbf{S}_1 \cup \mathbf{S}_2|}$

La figure 4.9 montre les performances obtenues pour différentes valeurs de seuil de similarité. L'indice de Jaccard est trop conservateur, car utilisé dans des environnements beaucoup plus petits, en *indoor* notamment. Il amène ainsi à ignorer un grand nombre de vraies libérations de parking, car l'environnement a changé entre-temps. Tanimoto et

ParkSense fonctionnent mieux, mais, même pour un coefficient idéal de 0,5, ils présentent un grand nombre de faux positifs, notamment car ils n'utilisent pas la puissance du signal reçu. Le coefficient de SmartPark, au contraire, permet d'identifier correctement 97% des libérations sans faux positif.

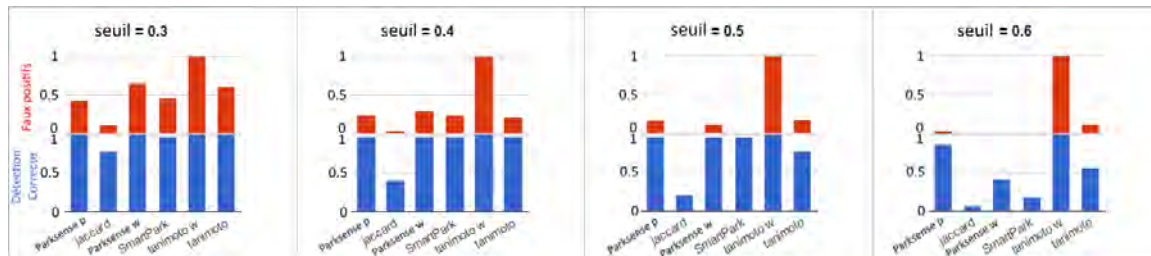


FIGURE 4.9 Taux de faux positifs et de détection pour différents coefficients de similarité et différents seuils.

4.3.4 Évaluation et conclusion

Nous avons demandé à sept volontaires dans trois villes différentes d'utiliser SmartPark pour une période de quelques mois. Nous leur avons demandé d'allumer leur smartphone et d'indiquer à l'application les moments où ils garaient et récupéraient leur véhicule. Cinq d'entre eux se déplacent plutôt en voiture tandis que deux autres prennent parfois un vélo. À la fin de l'expérience, nous avons obtenus 52 événements de parking et de libération dont 46 en voiture et 6 à vélo. SmartPark a réussi à chaque fois à détecter les événements de parking, avec un seul faux positif. Pour ce cas, l'utilisateur est allé dans la voiture d'un ami juste après s'être garé et à seulement quelques mètres. La figure 4.10 répertorie le pourcentage des modes de transport détectés pendant que le véhicule était garé. La figure 4.11 présente la fonction de distribution cumulative du temps de stationnement des utilisateurs.

Nous avons également voulu mesurer la consommation de batterie de SmartPark. Pour la limiter, SmartPark ne détecte pas en continu le mode de transport de l'utilisateur. Dans cette étude, l'estimation est effectuée toutes les 10s. C'est un compromis entre la rapidité de détection du changement de mode de transport et la consommation énergétique. Nous avons alors mesuré l'énergie consommée par les 6 capteurs. Nous avons trouvé $197mA$ en moyenne sur les sept smartphones de l'expérience. Cela représente $1615mJ$ toutes les 10s. SmartPark consomme également un scan passif à chaque vérification de localisation, qui représente environ $117mA$ pendant 1,07s soit $513mJ$ en moyenne. La figure 4.12 compare cette consommation à celle d'une application utilisant le GPS ou le Wi-Fi en

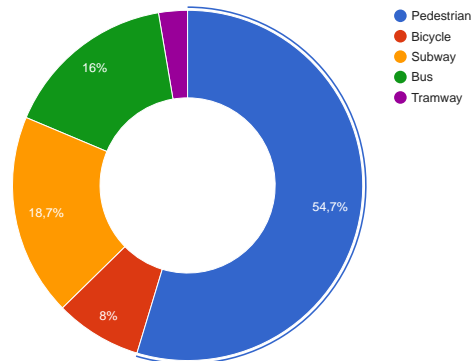


FIGURE 4.10 Pourcentage de détection du mode de transport pendant le stationnement.

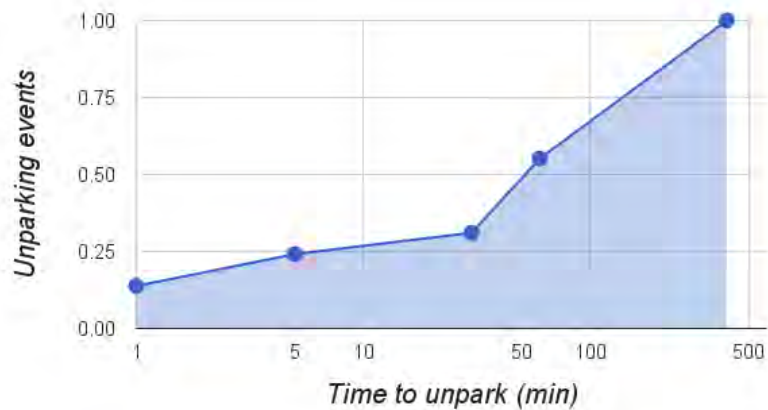


FIGURE 4.11 CDF du temps de stationnement.

continu. Pour obtenir cette courbe, nous avons simulé la consommation de batterie d'un smartphone LG Google Nexus 5 en utilisant les résultats obtenus sur deux heures de mesure réelles. On peut voir que SmartPark a un impact négligeable sur la batterie puisqu'au bout de 5h, la charge de la batterie n'a diminué que d'environ 4%, ce qui est beaucoup moins que pour les solutions Wi-Fi ou GPS.

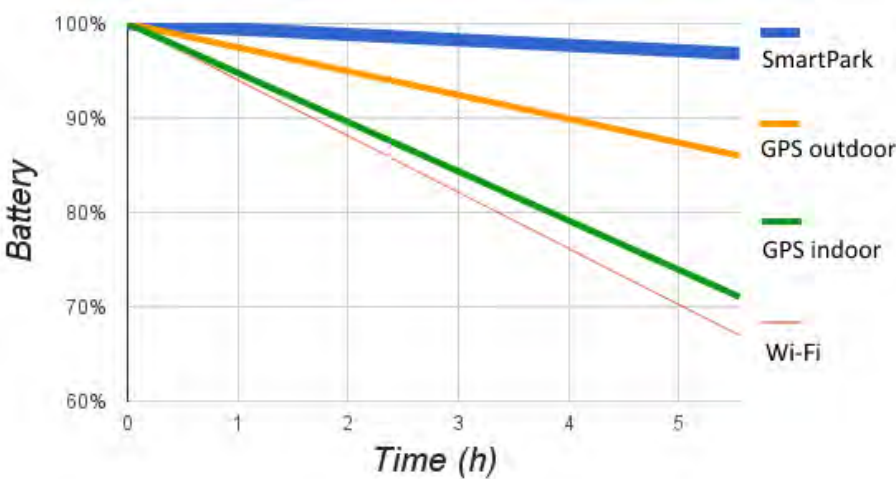


FIGURE 4.12 Consommation de batterie moyenne de SmartPark comparé au Wi-Fi et au GPS.

CONCLUSIONS ET PERSPECTIVES

Conclusions

Les technologies de l'information et de la communication (TIC) proposent des services de plus en plus nombreux aux usagers. Ces services se nourrissent des informations récoltées de l'utilisateur, et tout particulièrement sa localisation. Le smartphone joue un rôle majeur dans cette évolution car il accompagne son utilisateur dans tous ses déplacements. Si la navigation par satellite, et notamment la démocratisation du GPS, a résolu le problème de la localisation en extérieur, la localisation *indoor* reste un problème en suspens.

Bien qu'il existe des solutions depuis plusieurs années, elles sont toutes restreintes à un domaine d'application précis et ne permettent pas la navigation à l'intérieur d'un bâtiment inconnu. La trilatération du signal Wi-Fi permet une localisation imprécise, de l'ordre de la dizaine de mètres, rendant impossible la navigation. La prise d'empreintes Wi-Fi (ou Bluetooth) d'un bâtiment nécessite une phase de pré-déploiement et une connaissance du bâtiment. Les solutions UWB demandent l'installation de capteurs spécifiques dans le bâtiment et sur le smartphone de l'utilisateur. La dernière famille de solutions, fondée sur les capteurs de mouvement du smartphone, est grandement limitée par l'imprécision de ces capteurs.

Dans cette thèse, nous souhaitons obtenir une solution de navigation *indoor* qui permette à un utilisateur équipé de son smartphone de trouver son chemin dans un bâtiment inconnu, sans aucun besoin d'infrastructure ou de pré-déploiement. Pour cela, nous nous proposons d'étudier la faisabilité d'utiliser la navigation à l'estime à partir des capteurs de mouvement du smartphone, en dépit de leurs imprécisions de mesure.

Nous avons présenté Acrux, notre solution de localisation *indoor* qui repose sur la différenciation du mouvement humain en locomotions. Acrux permet de suivre le trajet de l'utilisateur en utilisant les données des accéléromètres du smartphone. Ceux-ci étant trop imprécis, Acrux n'utilise que leur forme pour identifier la locomotion de l'utilisateur et sa

cadence de marche. Acrux permet également de suivre les déplacements de l'utilisateur entre les différents étages d'un bâtiment lorsqu'il utilise des escaliers, un escalator ou un ascenseur. La précision d'Acrux, qui reste de l'ordre du mètre sur plusieurs centaines de mètres de déplacement, permet son utilisation dans une solution de navigation semblable à celles proposées en extérieur.

En reposant sur Acrux, ou sur une autre solution de localisation *indoor*, nous avons proposé InPreSS. InPreSS permet de recréer le plan d'un bâtiment grâce aux capteurs d'ambiance des smartphones des utilisateurs se déplaçant dans ce bâtiment. Ainsi, InPreSS peut proposer aux utilisateurs suivants de tracer leur trajectoire sur un plan dynamique. Par ailleurs, InPreSS permet également de calculer un chemin entre une source et une destination dans le bâtiment grâce à une représentation du bâtiment sous forme de graphe non-orienté valué. Nous avons notamment montré que 80 points de mesure permettent de retrouver tous les trajets possibles dans un bâtiment de $480m^2$.

Grâce à l'analyse des capteurs de mouvement d'InPreSS et à la détection de signatures de signaux d'Acrux, nous avons proposé d'étendre la détection de la locomotion humaine aux moyens de transports. Nous avons montré qu'il est possible de détecter le moyen de transport d'un utilisateur avec une précision de plus de 95% en utilisant une méthode de classification de type Random Forest. Nous avons alors proposé SmartPark, une application permettant de trouver des places de stationnement libres en ville, qui utilise cette détection du mode de transport.

Perspectives

Nous avons mis en évidence une solution de localisation et de création de plan qui, combinés, permettent la navigation dans un bâtiment inconnu, sans infrastructure ni pré-déploiement. Or, il existe des bâtiments connus, où il est tout à fait envisageable de réaliser un pré-déploiement ou d'ajouter une infrastructure. Par exemple, lors d'événements temporaires dans de grands espaces (festivals, conférences, etc.), les bâtiments sont isolés et une phase de prise d'empreintes peut être effectuée rapidement et sans gros surcoût. Dans ce cadre, et comme nous avons pu le présenter en introduction, il est tout à fait envisageable de combiner deux solutions telles que la navigation à l'estime proposée par Acrux et le *fingerprinting*. Cette dernière permet notamment d'obtenir une erreur de localisation qui ne dépend pas de la durée de déplacement, au contraire de la navigation à l'estime.

Par ailleurs, dans un cadre plus général, il est envisageable d'utiliser l'approche déterministe de la trilatération du signal Wi-Fi déjà existant pour résoudre le problème de la croissance de l'erreur d'Acrux au cours du temps et du déplacement.

Dans le cadre de cette thèse, nous avons également pu voir l'utilité des capteurs du smartphone dans les futures applications des technologies de l'information. Alors que nous les savions très imprécis, ils peuvent s'avérer suffisants dans certaines applications concrètes. Pour Acrux, c'est la compréhension de la locomotion humaine et son effet sur les capteurs de mouvement qui permet de corriger leur imprécision. Dans ce cadre, deux thématiques majeures semblent être des voies de recherche : la santé et les villes connectées.

Dans le domaine de la santé, de nombreux objets connectés sont aujourd'hui proposés dans le cadre de l'Internet des Objets. Pourtant, ces objets doivent être achetés séparément et sont parfois encombrants. Cela gêne énormément leur diffusion auprès du grand public. En revanche, les smartphones, même imprécis, peuvent fournir une première source d'information sur la santé de leurs utilisateurs, sans qu'ils aient à agir.

Par ailleurs, l'importance croissante de la thématique des *smart cities* a mené à envisager l'installation d'un très grand nombre de capteurs dans les rues et les bâtiments. Or, les smartphones offrent des capteurs embarqués, déjà connectés via le réseau téléphonique ou le Wi-Fi et qui suivent les mouvements de la population. De nombreuses applications liées aux villes intelligentes peuvent alors émerger à partir de ces informations. SmartPark, notre solution de gestion de stationnements intelligente s'inscrit dans ce cadre.

PUBLICATIONS

Conférences internationales

- J.-G. Krieg, G. Jakllari, H. Toma, and A.-L. Beylot, “Unlocking the smartphone’s senses for smart city parking,” in *IEEE ICC: International Conference on Communications*, 2016
- G. Artero Gallardo, J.-G. Krieg, G. Jakllari, L. Canourgues, and A.-L. Beylot, “COExiST: revisiting transmission count for cognitive radio networks,” in *ACM MSWiM: Proceedings of the 18th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2015

En cours de soumission

- J.-G. Krieg, G. Jakllari, H. Toma, and A.-L. Beylot, “Acrux: indoor localization without strings.” en cours de soumission
- J.-G. Krieg, G. Jakllari, and A.-L. Beylot, “InPRESS: indoor plan reconstruction using the smartphone’s senses.” soumis IEEE ICC ’17

Conférences francophones

- J.-G. Krieg, G. Jakllari, and A.-L. Beylot, “Smartpark : Se garer dans les smart cities ? un bon créneau pour les smartphones.” non-publié, 2016
- G. A. Gallardo, J.-G. Krieg, G. Jakllari, L. Canourgues, and A.-L. Beylot, “CO-ExiST: une métrique caractérisant la qualité des liens sans fil dans les réseaux de radios cognitives,” in *CoRes 2016: Rencontres Francophones sur la Conception de Protocoles, l'evaluation de Performance et l'Experimentation des Rseaux de Communication*, 2016

Brevets

- J.-G. Krieg, G. Jakllari, and A.-L. Beylot, “Acrux, solution de localisation indoor sur smartphone.” brevet en cours de dépôt, 2016

BIBLIOGRAPHIE

- [1] “Multi-trajets par Cisco.” <http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/82068-omni-vs-direct.html/>. Accessed : 2015-11-18.
- [2] “AWE communications.” <http://www.awe-communications.com/>. Accessed : 2016-11-02.
- [3] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, “Accurate gsm indoor localization,” in *ACM UbiComp : Ubiquitous Computing : 7th International Conference*, 2005.
- [4] C. Wu, Z. Yang, Y. Liu, and W. Xi, “WILL : wireless indoor localization without site survey,” *IEEE TPDS : Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.
- [5] “Angle of Arrival.” <http://www.apislabs.us/soter.html>. Accessed : 2016-11-02.
- [6] P. Lawitzki, *Application of Dynamic Binaural Signals in Acoustic Games*. Hochschule der Medien Stuttgart, 2012.
- [7] “Erreurs GPS.” <http://www.sxbluegps.com/fr/technologie/le-gps-et-les-erreurs/>. Accessed : 2015-12-21.
- [8] R. Wilson, *Propagation losses through common building materials*. Magis Networks, Inc., 2002.
- [9] “North Star Polaris.” <http://www.space.com/15567-north-star-polaris.html>. Accessed : 2016-08-30.
- [10] M. Schiffer, “The electric lighthouse in the nineteenth century : aid to navigation and political technology,” *Technology and Culture*, vol. 46, no. 2, pp. 275–305, 2005.
- [11] J. Bennett, *Divided Circle : A History of Instruments for Astronomy Navigation and Surveying*. Phaidon Inc Ltd, 1988.
- [12] B. W. Parkinson, T. Stansell, R. Beard, and K. Gromov, “A history of satellite navigation,” *Navigation*, vol. 42, no. 1, pp. 109–164, 1995.
- [13] F. C. Lane, *Venice, a maritime republic*. JHU Press, 1973.

- [14] P. Daly, "Navstar GPS and GLONASS : global satellite navigation systems," *IOSR JECE : Journal of Electronics and Communication Engineering*, vol. 5, pp. 349–357, 1993.
- [15] "Smartparking." <http://www.smartparking.com/>. Accessed : 2016-08-31.
- [16] "Smartqueue." <http://www.smartqueue.com.au/>. Accessed : 2016-08-31.
- [17] G.-Y. Jin, X.-Y. Lu, and M.-S. Park, "An indoor localization mechanism using active rfid tag," in *IEEE SUTC : International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006.
- [18] P. Bahl and V. N. Padmanabhan, "RADAR : an in-building rf-based user location and tracking system," in *IEEE InfoCom : 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000.
- [19] D. D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana, "Mobile ranging using low-accuracy clocks," *IEEE MTT : Transactions on Microwave Theory and Techniques*, vol. 48, no. 6, pp. 951–958, 2000.
- [20] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *IEEE InfoCom : 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [21] M. Youssef and A. Agrawala, "Handling samples correlation in the horus system," in *IEEE InfoCom : 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [22] "Polestar." <http://www.polestar.eu/fr/>. Accessed : 2016-07-20.
- [23] "Insiteo." <http://www.insiteo.com/joomla/index.php/fr/>. Accessed : 2016-07-27.
- [24] "Google Indoor Maps." <https://www.google.com/maps/about/partners/indoormaps/>. Accessed : 2016-08-27.
- [25] "Google maps." <https://www.google.fr/maps>. Accessed : 2016-08-31.
- [26] "Wordstream." <https://www.wordstream.com/geo-targeting>. Accessed : 2016-08-31.
- [27] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha, "PLEdestrians : a least-effort approach to crowd simulation," in *ACM SIGGRAPH : Proceedings of the Eurographics Symposium on Computer Animation*, 2010.
- [28] "Selecom." <http://www.selecom.fr/fr/couverture-indoor-batiments>. Accessed : 2016-08-31.
- [29] J. R. Blum, D. G. Greencorn, and J. R. Cooperstock, "Smartphone sensor reliability for augmented reality applications," *EAI MobiQuitous : Proceedings of the International Conference on Mobile and Ubiquitous Computing, Networking and Services*, 2012.

- [30] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping : part i," *IEEE RAM : Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [31] P. Laurent, G. Nard, and B. Manteaux, "SYLEDIS : a new concept for a radiopositioning system," in *OTC : Offshore Technology Conference*, 1974.
- [32] M. Bowman, S. K. Debray, and L. L. Peterson, "Reasoning about naming systems," *ACM TOPLAS : Transactions on Programming Languages and Systems*, vol. 15, no. 5, pp. 795–825, 1993.
- [33] B. Hofmann-Wellenhof, K. Legat, and M. Wieser, *Navigation : principles of positioning and guidance*. Springer Science & Business Media, 2011.
- [34] J. Freas and J. Lawrence, "Direct reading loran navigation system," 1951. US Patent 2,551,681.
- [35] M. Kayton and W. R. Fried, *Avionics navigation systems, 2nd Ed.* USA : John Wiley & Sons, 1997.
- [36] W. Guier and G. Weiffenbach, *Genesis of Satellite Navigation*. Johns Hopkins APL Digest, 1997.
- [37] P. Misra and P. Enge, *Radiowave propagation : Recommendation ITU-R P.1238-8*. ITU-R - International Telecommunication Union, 2015.
- [38] G. W. Hein, J. Godet, J.-L. Issler, J.-C. Martin, R. Lucas-Rodriguez, and T. Pratt, "Status of galileo frequency and signal design," in *CDROM : Proceedings on ION GPS*, pp. 34–37, 2002.
- [39] European Commission, "The galilei project - galileo design consolidation," 2004.
- [40] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi, "High-accuracy differential tracking of low-cost GPS receivers," in *ACM MobiSys : Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013.
- [41] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart, "COIN-GPS : indoor localization from direct gps receiving," in *ACM MobiSys : Proceedings of the 12th International Conference on Mobile Systems, Applications and Services*, 2014.
- [42] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *ACM MobiCom : Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015.
- [43] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993.
- [44] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "PinPoint : an asynchronous time-based location determination system," in *ACM MobiSys : Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, 2006.

- [45] S. A. Golden and S. S. Bateman, "Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging," *IEEE TMC : Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1185–1198, 2007.
- [46] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "SAIL : single access point-based indoor localization," in *ACM MobiSys : Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014.
- [47] D. Giustiniano, T. Bourchas, M. Bednarek, and V. Lenders, "Deep inspection of the noise in wifi time-of-flight echo techniques," in *ACM MSWiM : Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2015.
- [48] B. Sklar, *Digital communications : Fundamentals and Applications*, vol. 2. Prentice Hall NJ, 2001.
- [49] S. Yoon, K. Lee, and I. Rhee, "Fm-based indoor localization via automatic fingerprint db construction and matching," in *ACM MobiSys : Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013.
- [50] K. Liu, X. Liu, and X. Li, "Guoguo : enabling fine-grained indoor localization via smartphone," in *ACM MobiSys : Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013.
- [51] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM TOIS : Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.
- [52] P. Misra and P. Enge, *Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz*. Ganga-Jamuna Press, 2006.
- [53] J. Milanovic, S. Rimac-Drlje, and K. Bejuk, "Comparison of propagation models accuracy for wimax on 3.5 ghz," in *IEEE ICECS : 14th International Conference on Electronics, Circuits and Systems*, 2007.
- [54] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *ACM MobiCom : Proceedings of the 16th annual international conference on Mobile computing and networking*, 2010.
- [55] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee : zero-effort crowdsourcing for indoor localization," in *ACM MobiCom : Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012.
- [56] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *ACM MobiCom : Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, 2012.
- [57] J. Jun, Y. Gu, L. Cheng, B. Lu, J. Sun, T. Zhu, and J. Niu, "Social-Loc : improving indoor localization with social sensing," in *ACM SenSys : Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013.

- [58] A. Papapostolou and H. Chaouchi, "Wife : wireless indoor positioning based on fingerprint evaluation," in *IFIP Networking : International Conference on Research in Networking*, 2009.
- [59] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE SMC : Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 580–585, 1985.
- [60] M. Ficco, C. Esposito, and A. Napolitano, "Calibrating indoor positioning systems with low efforts," *IEEE TMC : Transactions on Mobile Computing*, vol. 13, no. 4, pp. 737–751, 2014.
- [61] X. Chai and Q. Yang, "Reducing the calibration effort for probabilistic indoor location estimation," *IEEE TMC : Transactions on Mobile Computing*, vol. 6, no. 6, pp. 649–662, 2007.
- [62] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *ACM MobiSys : Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, 2005.
- [63] W. S. A. Au, C. Feng, S. Valaee, S. Reyes, S. Sorour, S. N. Markowitz, D. Gold, K. Gordon, and M. Eizenman, "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE TMC : Transactions on Mobile Computing*, vol. 12, pp. 2050–2062, 2013.
- [64] A. S. Paul and E. A. Wan, "Wi-fi based indoor localization and tracking using sigma-point kalman filtering methods," in *IEEE/ION PLANS : Position, Location and Navigation Symposium*, 2008.
- [65] "BeSpoon." <http://bespoon.com/>. Accessed : 2016-07-17.
- [66] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *ACM MobiSys : Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013.
- [67] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *ACM MobiCom : Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [68] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, and X. Li, "Towards ubiquitous indoor localization service leveraging environmental physical features," in *IEEE InfoCom : International Conference on Computer Communications*, 2014.
- [69] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense : mobile phone localization via ambience fingerprinting," in *ACM MobiCom : Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, 2009.
- [70] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No Need to War-drive : unsupervised indoor localization," in *ACM MobiSys : Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, 2012.

- [71] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *ACM UbiComp : Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012.
- [72] M. M. Samson, A. Crowe, P. L. de Vreede, J. A. G. Dessens, S. A. Duursma, and H. J. J. Verhaar, "Differences in gait parameters at a preferred walking speed in healthy subjects due to age, height and body weight," *Aging Clinical and Experimental Research*, vol. 13, no. 1, pp. 16–21, 2001.
- [73] A. Kuo, "A simple model of bipedal walking predicts the preferred speed–step length relationship," in *ASME Journal of Biomechanical Engineering*, pp. 264–269, 2001.
- [74] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *ACM MobiSys : Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014.
- [75] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose : indoor positioning with mobile phones and visible light," in *ACM MobiCom : Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [76] L. Li, C. Zhao, F. Zhao, and J. Shen, "Travi-Navi : self-deployable indoor navigation system," in *ACM MobiCom : Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [77] "Locomotion wikipedia." <https://fr.wikipedia.org/wiki/Locomotion>. Accessed : 2016-09-28.
- [78] R. Alexander, "Optimization and gaits in the locomotion of vertebrates," *Physiological Reviews*, 1989.
- [79] "Android SensorEvent." <https://developer.android.com/reference/android/hardware/SensorEvent.html>. Accessed : 2016-11-03.
- [80] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona Lisa : spot localization using phy layer information," in *ACM MobiSys : Proceedings of the 10th International Conference on Mobile Systems, Applications and Services*, 2012.
- [81] S. He, T. Hu, and S.-H. G. Chan, "Contour-based trilateration for indoor fingerprinting localization," in *ACM SenSys : Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015.
- [82] J. Xiong and K. Jamieson, "ArrayTrack : a fine-grained indoor location system," in *USENIX NSDI : Proceedings of the 10th Symposium on Networked Systems Design and Implementation*, 2013.
- [83] C. Zhang, F. Li, J. Luo, and Y. He, "iLocScan : harnessing multipath for simultaneous indoor source localization and space scanning," in *ACM SenSys : Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems*, 2014.

- [84] M. Denny, *The Science of navigation : From Dead Reckoning to GPS*. Johns Hopkins University Press, 2012.
- [85] J. Georgy, A. Noureldin, M. J. Korenberg, and M. M. Bayoumi, “Modeling the stochastic drift of a MEMS-based gyroscope in gyro/odometer/GPS integrated navigation,” *IEEE ITS : Transactions on Intelligent Transportation System*, vol. 11, no. 4, pp. 856–872, 2010.
- [86] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, “Did you see Bob ? : human localization using mobile phones,” in *ACM MobiCom : Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*, 2010.
- [87] I. Constandache, R. Choudhury, and I. Rhee, “Towards mobile phone localization without war-driving,” in *IEEE InfoCom : International Conference on Computer Communications*, 2010.
- [88] A. Anjum and I. M.U., “Activity recognition using smartphone sensors,” in *IEEE CCNC : Consumer Communications and Networking Conference*, 2013.
- [89] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, “Robust indoor positioning with lifelong learning,” *IEEE JSAC : Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2287–2301, 2015.
- [90] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in *ACM SenSys : Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013.
- [91] P. Zhou, M. Li, and G. Shen, “Use it free : instantly knowing your phone attitude,” in *ACM MobiCom : Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [92] D. R. MacNulty, L. D. Mech, and D. W. Smith, “A proposed ethogram of large-carnivore predatory behavior, exemplified by the wolf,” *Journal of Mammalogy*, vol. 88, no. 3, pp. 595–605, 2007.
- [93] E. Hiris, A. Krebeck, J. Edmonds, and A. Stout, “What learning to see arbitrary motion tells us about biological motion perception.,” *Journal of Experimental Psychology : Human Perception and Performance*, vol. 31, no. 5, p. 1096, 2005.
- [94] S. W. Lipfert, M. Günther, D. Renjewski, S. Grimmer, and A. Seyfarth, “A model-experiment comparison of system dynamics for human walking and running,” *Journal of Theoretical Biology*, vol. 292, pp. 11–17, 2012.
- [95] A. Thorstensson and H. Roberthson, “Adaptations to changing speed in human locomotion : speed of transition between walking and running,” *Acta Physiologica Scandinavica*, vol. 131, no. 2, pp. 211–214, 1987.
- [96] G. Welk, J. Differding, R. Thompson, S. Blair, J. Dziura, and P. Hart, “The utility of the digi-walker step counter to assess daily physical activity patterns,” *ACSM MMSE : Medicine and Science in Sports and Exercise*, 2000.

- [97] H. Hasegawa, T. Yamauchi, and W. J. Kraemer, "Foot strike patterns of runners at the 15-km point during an elite-level half marathon," *NSCA JSCR : Journal of Strength and Conditioning Research*, vol. 21, no. 3, 2007.
- [98] G. Danielson and C. Laczos, "Some improvements in practical fourier analysis and their application to x-ray scattering from liquids," in *Journal of the Franklin Institute*, 1942.
- [99] O. J. Woodman, *An introduction to inertial navigation*. University of Cambridge Technical Report, 2007.
- [100] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors : more hype than hope!," in *ACM HotMobile : Proceedings of the 17th Workshop on Mobile Computing Systems and Applications*, 2014.
- [101] H. Ye, T. Gu, X. Tao, and J. Lu, "B-Loc : scalable floor localization using barometer on smartphone," in *IEEE MASS : Proceedings of the 11th International Conference on Mobile Ad-Hoc and Sensor Systems*, 2014.
- [102] "Escalator." http://en.wikipedia.org/wiki/Escalator\#Operation_and_layout. Accessed : 2016-09-27.
- [103] "Android app power tutor." <https://powertutor.org>. Accessed : 2016-09-28.
- [104] E. Gibbon, *The history of the decline and fall of the Roman Empire*, vol. 3. Harper & Brothers, 1841.
- [105] "Yahoo maps." <https://maps.yahoo.com/>. Accessed : 2016-10-10.
- [106] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw : Indoor floor plan reconstruction via mobile crowdsensing," in *ACM MobiCom : Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014.
- [107] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM : A factored solution to the simultaneous localization and mapping problem," in *AAAI : Proceedings of the National Conference on Artificial Intelligence*, 2002.
- [108] F. Adib and D. Katabi, "See through walls with wifi!," in *ACM SigComm : International Conference on the Applications, Technologies, Architectures, and Protocols for Computer Communication*.
- [109] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor wifi graphslam," in *IEEE ICRA : International Conference on Robotics and Automation*, 2011.
- [110] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE TMC : Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2015.
- [111] D. Reem, "An algorithm for computing voronoi diagrams of general generators in general normed spaces," in *IEEE ISVD : 6th International Symposium on Voronoi Diagrams*, 2009.

- [112] J. Wang and W. J. Rugh, "On the pseudo-linearization problem for nonlinear systems," *Systems & Control Letters*, vol. 12, no. 2, pp. 161–167, 1989.
- [113] J. A. Hartigan and M. A. Wong, "Algorithm AS 136 : a k-means clustering algorithm," *Journal of the Royal Statistical Society*, vol. 28, pp. 100–108, 1979.
- [114] G. J. Szekely and M. L. Rizzo, "Hierarchical clustering via joint between-within distances : extending ward's minimum variance method," *Journal of Classification*, vol. 22, no. 2, pp. 151–183, 2005.
- [115] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [116] E. W. Dijkstra, *A short introduction to the art of programming*, vol. 4. Technische Hogeschool Eindhoven, 1971.
- [117] L. Breiman, "Random forests," *Journal of Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [118] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
- [119] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, vol. 3001 of *Lecture Notes in Computer Science*, pp. 1–17, 2004.
- [120] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay, "Activity sensing in the wild : a field trial of ubifit garden," in *ACM SigChi : Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.
- [121] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in *ACM UbiComp 2008 : 10th International Conference on Ubiquitous Computing*.
- [122] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on GPS data for web applications," *ACM TWEB : Transactions on the Web*, 2010.
- [123] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gis information," in *ACM SIGSPATIAL : Proceedings of the 19th International Conference on Advances in Geographic Information Systems*, 2011.
- [124] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse gps data using a moving window svm classification," vol. 36, pp. 526–537, 2012.
- [125] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM TOSN : Transactions on Sensor Networks*, vol. 6, no. 2, pp. 13 :1–13 :27, 2010.

- [126] H. J. T. Tao Feng, "Transportation mode recognition using gps and accelerometer data," *Transportation Research Part C : Emerging Technologies*, pp. 118–130, 2013.
- [127] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga, "Sensing motion using spectral and spatial analysis of wlan rssi," in *EuroSSC : Proceedings of the 2nd European Conference on Smart Sensing and Context*, 2007.
- [128] J. Krumm and E. Horvitz, "LOCADIO : inferring motion and location from wi-fi signal strengths," in *EAI MobiQuitous : Proceedings of the International Conference on Mobile and Ubiquitous Computing, Networking and Services*, 2004.
- [129] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. de Lara, *Mobility Detection Using Everyday GSM Traces*. 2006.
- [130] J. B. M. Mun, D. Estrin and M. Hansen, "Parsimonious mobility classification using gsm and wifi traces," in *ACM SenSys : Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, 2008.
- [131] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," in *IEEE APWCS : Proceedings of the Asia-Pacific Conference on Wearable Computing Systems*, 2010.
- [132] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *ACM SenSys : Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013.
- [133] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *PERVASIVE : Proceedings of the 4th International Conference on Pervasive Computing*, 2006.
- [134] W. H, "Partial least squares," *Encyclopedia of Statistical Sciences*, 2006.
- [135] R. D. Tobias, "An introduction to partial least squares regression," 1995.
- [136] I. Cramer, Richard D., "Partial least squares (PLS) : its strengths and limitations," *Perspectives in Drug Discovery and Design*, vol. 1, no. 2, pp. 269–278, 1993.
- [137] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *IEEE ICCV : 12th International Conference on Computer Vision Workshops*, 2009.
- [138] Urbitran/Parsons Brinckerhoff, Eng-Wong, Taub & Associates, Howard/Stein-Hudson Associates, Inc., HOK Planning, "Parking study : Summary of Typical User Characteristics," 2008.
- [139] S. Nawaz, C. Efstratiou, and C. Mascolo, "ParkSense : a smartphone based sensing system for on-street parking," in *ACM MobiCom : Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, 2013.
- [140] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

- [141] J. Hanselman, "Cincinnati has lots of unpaid parking tickets," *WVXU*, 2014.
- [142] "Portland bureau of transportation." <https://www.portlandoregon.gov/transportation/article/99190/>. Accessed : 2015-10-30.
- [143] F. O'Sullivan, "Amsterdam has officially run out of spaces to park its bicycles," *Citylab*, 2015.
- [144] "Google open spot : a useful application that no one uses." <http://www.androidauthority.com/googlelabs-open-spot-a-useful-application-that-noone-uses-15186/>. Accessed : 2015-10-30.
- [145] "SFpark." <http://www.sfpark.org/>. Accessed : 2015-10-30.
- [146] U.S. Department of Transportation , "A cross-cutting study, taking the stress out of parking," *Advanced Parking Management Systems*, 2007.
- [147] M. Kjaergaard, M. Wirz, D. Roggen, and G. Troster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *IEEE PerCom : International Conference on Pervasive Computing and Communications*, 2012.
- [148] J.-G. Krieg, G. Jakllari, H. Toma, and A.-L. Beylot, "Unlocking the smartphone's senses for smart city parking," in *IEEE ICC : International Conference on Communications*, 2016.
- [149] G. Artero Gallardo, J.-G. Krieg, G. Jakllari, L. Canourgues, and A.-L. Beylot, "COExiST : revisiting transmission count for cognitive radio networks," in *ACM MSWiM : Proceedings of the 18th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2015.
- [150] J.-G. Krieg, G. Jakllari, H. Toma, and A.-L. Beylot, "Acrux : indoor localization without strings." en cours de soumission.
- [151] J.-G. Krieg, G. Jakllari, and A.-L. Beylot, "InPreSS : indoor plan reconstruction using the smartphone's senses." soumis à IEEE ICC '17.
- [152] J.-G. Krieg, G. Jakllari, and A.-L. Beylot, "Smartpark : Se garer dans les smart cities? un bon créneau pour les smartphones." non-publié, 2016.
- [153] G. A. Gallardo, J.-G. Krieg, G. Jakllari, L. Canourgues, and A.-L. Beylot, "CO-ExiST : une métrique caractérisant la qualité des liens sans fil dans les réseaux de radios cognitives," in *CoRes 2016 : Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, 2016.
- [154] J.-G. Krieg, G. Jakllari, and A.-L. Beylot, "Acrux, solution de localisation indoor sur smartphone." brevet en cours de dépôt, 2016.

N'hésitez pas à aller sur la plage si vous voulez faire du bruit.
un inconnu de Guidel-Plage

Résumé

L'environnement *indoor* permet un grand nombre de services issus des technologies de l'information et de la communication. La localisation de l'utilisateur, via celle de son smartphone, est donc un élément-clé de cette réussite. Cette thèse s'intéresse au suivi des déplacements de l'utilisateur grâce aux capteurs de mouvement embarqués dans son smartphone. Elle repose sur la détermination du type de locomotion. Nous proposons une solution de navigation *indoor* complète, permettant de proposer à l'utilisateur un chemin jusqu'à sa destination dans n'importe quel bâtiment tout en connaissant sa position à chaque instant, avec une précision de l'ordre du mètre. De façon analogue, nous avons également montré que nous pouvons déterminer le mode de transport d'un utilisateur pour une application de détection de places de parking libres.

Mots clés : Localisation, smartphones, capteurs, accéléromètres, *indoor*, intérieur, navigation.

Abstract

Indoor environments present opportunities for a rich set of location-aware services in the information and communications technology (ICT) area. Therefore, accurately localizing a user indoors has become a key enabling technology. This thesis addresses the issue of tracking a user equipped with an off-the-shelf smartphone by exploiting its embedded motion sensors. Leveraging key characteristics of human locomotion, we propose a complete, infrastructure-free indoor navigation solution, allowing a user to navigate any unknown building with meter-level accuracy. Finally, extending our understanding of locomotion to outdoors areas where users are inside vehicles, we design and implement a smartphone application for smart on-street parking.

Keywords : Indoor localization, smartphones, motion sensors, navigation, on-street parking