



HAL
open science

Méthodes de localisation de capteurs dans le contexte de l'Internet des Objets

Wafa Njima

► **To cite this version:**

Wafa Njima. Méthodes de localisation de capteurs dans le contexte de l'Internet des Objets. Traitement des images [eess.IV]. Conservatoire national des arts et métiers - CNAM; École supérieure des communications de Tunis (Tunisie), 2019. Français. NNT : 2019CNAM1264 . tel-02484757

HAL Id: tel-02484757

<https://theses.hal.science/tel-02484757v1>

Submitted on 19 Feb 2020

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.

**ÉCOLE DOCTORALE D'INFORMATIQUE, TELECOMMUNICATIONS ET
ELECTRONIQUE**

Centre d'études et de recherche en informatique et communications

En cotutelle avec

**ÉCOLE DOCTORALE EN TECHNOLOGIES DE L'INFORMATION ET DE LA
COMMUNICATION**

Innovation of communicant and cooperative mobiles

THÈSE

Présentée par :

Wafa NJIMA

Soutenue le : **13 Décembre 2019**

Pour obtenir le grade de :

Docteur du Conservatoire National des Arts et Métiers

Discipline / Spécialité : **Génie informatique, automatique et traitement du signal /
Radiocommunications**

ET

Docteur de l'Ecole Supérieure de Communications de Tunis

Spécialité : **Technologies de l'information et de la communication**

**Méthodes de localisation de capteurs dans le contexte
de l'Internet des Objets**

THÈSE dirigée par :

Prof. Michel TERRE

Prof. Ridha BOUALLEGUE

CNAM Paris, France

Sup'COM Tunis, Tunisie

Et co-encadrée par :

Dr. Rafik ZAYANI

Dr. Iness Ahriz

ISI Tunis, Tunisie

CNAM Paris, France

RAPPORTEURS :

Prof. Geneviève BAUDOIN

Prof. Tahar EZZEDINE

ESIEE Paris, France

ENIT Tunis, Tunisie

EXAMINATEURS :

Prof. Maryline HELARD

Prof. Jamel BELHADJ TAHER

Insa Rennes, France

ENISo Sousse, Tunisie

Remerciements

Avant tout, je voudrais exprimer ma plus profonde gratitude à mes directeurs de thèse Prof. Michel TERRE et Prof. Ridha BOUALLEGUE pour leurs encadrements de qualité, leurs encouragements et leurs précieux conseils tout au long de cette thèse. Je remercie également Dr. Iness AHRIZ et Dr. Rafik ZAYANI pour avoir co-encadré mes travaux. Un grand merci pour leurs soutiens, leurs disponibilités et leur aide à surmonter les difficultés tout au long de cette thèse. J'ai beaucoup bénéficié de nos discussions. J'exprime ma profonde gratitude aux membres du jury, pour l'honneur qu'ils me font en acceptant de juger ce travail et d'examiner les résultats de mes recherches. De plus, je tiens à remercier toute l'équipe du centre d'études et de recherche en informatique et communications CEDRIC du CNAM ainsi que mes collègues au laboratoire de recherche Innov'COM, c'est très agréable de travailler avec eux. Je tiens également à remercier tous mes amis pour leur aide et leurs soins afin que cette thèse soit accomplie. Enfin, je voudrais remercier ma famille pour leur amour inconditionnel et leur soutien sans fin. Mes remerciements vont naturellement à mes parents pour leur confiance, leurs sacrifices et leurs encouragements dans les moments les plus difficiles malgré la distance qui me sépare d'eux. Mes remerciements vont également à mes chères soeurs, à mon beau frère, à mes nièces et à mon neveu qui m'ont toujours supporté et cru en moi. Cette thèse est dédiée à vous.

Paris, Décembre 2019

Wafa NJIMA

Abstract

With the growing emergence of the Internet of Things and the importance of position information in this context, localization is attracting more and more attention in the researchers' community. The outdoor location is provided by the Global Positioning System (GPS) which is not suitable for indoor environments. Several indoor localization techniques exist, but no standard has been developed. Existing methods are mainly based on trilateration or fingerprinting. Trilateration is a geometric method that exploits the distances between an object and reference points to locate it. Here, we consider the case where some small number of sensors (anchor nodes) whose coordinates are known by installing them at known points, and the rest unknown nodes, have to determine their own coordinates using measured inter-sensor distances and anchor nodes coordinates. Since, in IoT, sensors are not capable of high-power transmission which would be unable to make measurements with all nodes. Therefore, each unknown node has available only the distance measurements to a few number of neighboring nodes. Consequently, the trilateration would not be able to offer good performance or even more could not work. Then, in this investigation, we propose to enhance the trilateration by combining it with a technique that aims at completing the inter-node distances matrix from the partially known data. This approach is known as 'matrix completion'. Specifically, the proposed classification scheme is formulated as a simple optimization problem which enables efficient and reliable algorithm implementation. The developed algorithm is based on the steepest Gradient Descent (GD) associated to many advanced optimization methods that are deployed and compared in terms of complexity and localization accuracy in order to identify the most adequate for the given problem. On another hand, the fingerprinting approach shown to offer very good localization accuracy. This method compares the fingerprint associated to the device to be located to a fingerprints' database constructed in offline. In this thesis, a comparative study between different metrics of similarity evaluation has been conducted. We have demonstrated that these metrics can provide very good performance but in the cost of a very high computational complexity which makes it not adequate for a localization in real time. In order to overcome this major problem, we have developed a localization framework that shifts the online prediction complexity to an offline preprocessing step, based on a linear model or a Convolutional Neural Network (CNN) model. The development of a linear model generates a mathematical relation that links the powers of the signal received by an object and its coordinates. When using a CNN model, the localization problem is formulated as radio image-related region classification. The results demonstrate the outperformance of the developed CNN, in terms of localization accuracy and complexity compared to other classical approaches.

Keywords : Indoor localization, RSSI, trilateration, fingerprints, matrix completion, convolutional neural networks (CNN).

Résumé

Grâce à l'émergence croissante de l'Internet des Objets et à l'importance de l'information de position dans ce contexte, la localisation attire de plus en plus d'attention dans la communauté des chercheurs. La localisation en extérieur est assurée par le GPS qui n'est pas adapté aux environnements intérieurs. Plusieurs techniques de localisation en intérieur existent, mais il n'existe pas encore un standard. L'objectif de cette thèse est d'améliorer les techniques de localisation existantes tout en maintenant un niveau de localisation satisfaisant avec une faible complexité de calcul. Afin de surmonter les inconvénients des techniques de localisation existantes, nous avons proposé des approches alternatives. Dans un contexte de communication distribuée, la trilatération a été combinée avec un processus d'optimisation qui vise à compléter la matrice de distances inter noeuds à partir des données partiellement connues. Des algorithmes d'optimisation avancés ont été utilisés en développant la solution analytique correspondante à chacun. Ainsi une solution de localisation pour une architecture distribuée a été proposée. Pour une architecture centralisée utilisant la technique du fingerprinting contenant les puissances reçues, plusieurs stratégies ont été étudiées. Une étude comparative entre les différentes métriques d'évaluation de similarité a été développée. Cette étude a été suivie par le développement d'un modèle linéaire entre le fingerprint de test et les fingerprints d'une base de données générant une relation mathématique qui relie les puissances du signal reçues par un objet à ses coordonnées. Cela aide à diminuer la complexité de calcul en ligne et ainsi mieux s'adapter à un système temps réel. Enfin, la relation entre les puissances reçues et les coordonnées a été confiée à un réseau de neurones convolutif (CNN) qui traite le problème de localisation comme un problème de classification d'images radio. Les performances de toutes les approches proposées ont été évaluées et discutées. Ces résultats montrent bien l'amélioration des performances des techniques basiques en termes de précision de localisation et de complexité.

Mots clés : La localisation en intérieur, la puissance du signal reçu, la trilatération, le fingerprinting, la complétion de matrice, les réseaux de neurones convolutifs (CNN).

Table des matières

Remerciements	1
Abstract	3
Résumé	5
Liste des figures	9
Liste des tableaux	11
Glossaire	13
Introduction	17
1 La localisation indoor : Objectifs, applications et techniques	23
1.1 Introduction	23
1.2 L'Internet des Objets (IoT)	24
1.3 Applications diverses de l'IoT liées à la localisation	25
1.4 Les technologies de communication utilisées pour des fins de localisation en intérieur	27
1.4.1 Technologies émergentes exploitées dans la localisation	27
1.4.2 Les technologies de communication radio	29
1.5 Les paramètres du signal radio	31
1.5.1 L'angle d'arrivée (AoA)	31
1.5.2 Temps de propagation des signaux	32
1.5.3 La réponse impulsionnelle du canal (CIR)	33
1.5.4 Phase d'arrivée (PoA)	33
1.5.5 La puissance reçue du signal	33
1.6 Les techniques de localisation	34
1.6.1 Proximité	34
1.6.2 Les méthodes géométriques	34
1.6.3 La méthode de fingerprinting	38
1.7 Les critères d'évaluation de performances	41
1.8 Objectifs de la thèse	42
1.9 Conclusion	42
2 Solution de localisation pour une architecture distribuée	43
2.1 Système modèle	43
2.2 Localisation à base de la complétion de matrice	45
2.2.1 Formulation du problème de la complétion de matrice	45

2.2.2	Complétion de matrice : Optimisation avec Gradient Descendant (GD) et ses variantes	47
2.2.3	Localisation à base de trilatération	50
2.3	Evaluation des performances	52
2.3.1	Configuration de l'environnement de simulation	52
2.3.2	Vérification de la propriété du rang faible	53
2.3.3	Performances de récupération et précision de localisation	53
2.3.4	Etude de complexité	57
2.4	Vérification de l'algorithme en utilisant une base de mesures réelles	60
2.5	Conclusion	61
3	Solutions de localisation pour une architecture centralisée	63
3.1	Système modèle	64
3.2	Evaluation de similarité	64
3.2.1	Métriques d'évaluation	65
3.2.2	Estimation de la position	66
3.2.3	Présentation des données	67
3.2.4	Présentation et discussion des résultats et évaluation de performances	67
3.3	Sélection de PAs	71
3.3.1	Présentation et discussion des résultats	73
3.4	Localisation à base d'un système analytique	77
3.4.1	Développement de Taylor pour une localisation en intérieur	77
3.4.2	Evaluation des performances et validation du système	80
3.5	Localisation avec un modèle à base des réseaux de neurones convolutifs CNN	85
3.5.1	Application du Deep CNN dans la localisation en intérieur	85
3.5.2	Les données	85
3.5.3	Les modèles de réseaux de neurones convolutifs	89
3.5.4	Résultats : Evaluation des performances du Deep CNN	91
3.6	Conclusion	98
	Conclusions et perspectives	99
	Bibliographie	101

Table des figures

1.1	Les types d'architecture en IoT.	25
1.2	Principe de l'AoA dans la localisation.	32
1.3	Principe de la trilatération.	35
1.4	Principe de fonctionnement du GPS (Source : https://www.geotracteur.fr/).	35
1.5	Localisation avec AoA -Triangulation-.	38
1.6	Les étapes de la technique de fingerprinting en considérant N points de référence et M positions d'apprentissage.	39
2.1	Les détails du framework développé à base de la complétion de matrice en considérant un réseau contenant 7 noeuds capteurs dont 3 sont des anchors.	44
2.2	Configuration du réseau de capteurs sans fil.	52
2.3	Fraction d'énergie capturée par les valeurs singulières.	54
2.4	Performances des algorithmes avec sigma shadowing = 0.	55
2.5	Performances des algorithmes avec sigma shadowing = 2.	55
2.6	Performances des algorithmes avec sigma shadowing = 5.	56
2.7	Configuration utilisée lors des expérimentations réelles.	60
2.8	CDF avec des mesures réelles.	61
3.1	Exemple d'architecture avec 5 PAs et 12 positions de mesures.	64
3.2	La valeur d'entropie correspondant à chaque PA détecté au niveau de l'étage 3 du bâtiment 0.	71
3.3	Les étapes de l'algorithme de localisation assurant une sélection des PAs.	72
3.4	CDFs de différents étages du bâtiment 0.	74
3.5	CDFs de différents étages du bâtiment 1.	75
3.6	CDFs de différents étages du bâtiment 2.	75
3.7	Scénario d'une configuration en intérieur avec 6 PAs et 25 points d'apprentissage.	81
3.8	CDF de l'erreur de localisation avec 1% d'erreur sur les puissances reçues pour 25 points d'apprentissage.	82
3.9	CDF de l'erreur de localisation avec 2% d'erreur sur les puissances reçues pour 25 points d'apprentissage.	82
3.10	CDF de l'erreur de localisation avec 5% d'erreur sur les puissances reçues pour 25 points d'apprentissage.	83
3.11	CDF de l'erreur de localisation pour $\gamma = 3$ avec 25 points d'apprentissage et des PAs placés à 1000 m ou 2000 m du centre de l'espace intérieur.	84
3.12	Les différentes étapes de localisation à base du CNNLocWK.	86

3.13	Partition de l'environnement intérieur en zones.	87
3.14	La structure des bases RSSI pour chaque point d'apprentissage.	87
3.15	La structure des images radio.	88
3.16	Un exemple d'architecture d'un réseau CNN ayant 2 couches de convolution et 1 couche entièrement connectée.	89
3.17	L'opération de max-pooling avec des images radio (fenêtre de taille (2×2) et un pas de 2).	90
3.18	Variation de la précision en fonction du paramètre T	93

Liste des tableaux

2.1	L'algorithme décrit en 4 étapes.	46
2.2	Les valeurs de différents paramètres utilisés.	57
2.3	La complexité des algorithmes normalisée par rapport à 'GD'.	59
3.1	Description des champs d'un fingerprint de la base UJIndoorLoc.	68
3.2	L'erreur moyenne de localisation en mètres correspondante à chaque combinaison.	69
3.3	Tableau comparatif des approches utilisant des mesures réelles de RSSI.	70
3.4	Le nombre de PAs sélectionné correspondant à chaque phase de sélection.	74
3.5	L'erreur moyenne en mètres pour chaque étage.	76
3.6	L'erreur médiane en mètres pour chaque étage.	76
3.7	Description de l'algorithme de localisation avec un développement de Taylor de l'ordre 2.	80
3.8	Erreur de localisation en mètres pour 90% de points avec $N = 25$ et $N = 9$ pour différentes valeurs d'erreurs sur les puissances moyennes reçues.	84
3.9	Liste des hyperparamètres considérés.	92
3.10	Variation de la précision en fonction de T en utilisant une grille de dimension $5\text{ m} \times 5\text{ m}$	94
3.11	Variation de la précision en fonction de T en utilisant une grille de dimension $2\text{ m} \times 2\text{ m}$	94
3.12	La précision d'ADAM avec les données de validation en utilisant des grilles de dimension $2\text{ m} \times 2\text{ m}$ et 10 PAs.	95
3.13	Variation de la précision en fonction du nombre de couches en utilisant une grille de dimension $2\text{ m} \times 2\text{ m}$ et 10 PAs.	96
3.14	Les architectures des modèles de deep learning utilisés.	97
3.15	Comparison de la précision associée aux différents algorithmes en utilisant des grilles de dimension $2\text{ m} \times 2\text{ m}$ et 10 PAs.	97

Glossaire

Abréviations et Acronymes

Adagrad	Adaptative Gradient
Adam	Adaptative Moment Estimation
ALM	Augmented Lagrange Multiplier
AoA	Angle d'arrivée
APG	Approximal Gradient
BLE	Bluetooth
CDF	Cumulative Distribution Function
CIR	Réponse Impulsionnelle du Canal
CNN	Réseaux de neurones convolutifs
CS	Compressed Sensing
CSI	Channel State Information
DL	Deep Learning
DNN	Réseaux de neurones profonds
FC	Fully Connected
GD	Gradient Descendant
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IoT	Internet of Things
KNN	K-Nearest Neighbors
LED	Light Emitting Diode
LOS	Line Of Sight
MC	Complétion de matrice
MDE	Matrice de Distances Euclidiennes
MDS	Positionnement Multidimensionnel
ML	Machine Learning
MLP	Multi-layer Perceptron
MSE	Mean Square Error
NAG	Nesterov Accelerated Gradient
NLOS	Non Line Of Sight
NN	Réseaux de neurones
PA	Point d'accès
PC	Composants Principaux

PoA	Phase of Arrival
PR	Point de référence
RFID	Radio Frequency Identification
RMSProp	Root Mean Square Propagation
RNN	Réseaux de neurones récurrents
RSD	Relaxation Semi-Définie
RSSI	Received Signal Strength Indicator
RToF	Round-Trip Time of Flight
SDP	Semi Défini Positif
SVD	Décomposition en Valeurs Singulières
SVT	Singular Value Thresholding
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
UIT	Union Internationale de Communications
UMF	Unimodality constrained Matrix Factorization
UWB	Ultra Wideband
VLC	Communication par lumière visible
WiFi	Réseau local sans fil

Notations

Les matrices sont présentées par des lettres majuscules en gras. Les autres notations sont résumées comme suit :

$[\cdot]^T$	Opérateur transposé.
$ x $	Valeur absolue d'un scalaire.
\mathbf{X}^{-1}	Inverse de la matrice \mathbf{X} .
\mathbf{I}_n	Matrice identité de dimension $(n \times n)$.
$\ \cdot\ _2$	Norme 2.
$\ \cdot\ _F$	Norme Frobenius.
$\ \cdot\ _*$	Norme Nucléaire.
$\mathbf{X}^{(t)}$	Matrice obtenue à l'itération t .
$\mathbf{X} \cdot \mathbf{Y}$	Produit des matrices.
$\mathbf{X} \odot \mathbf{Y}$	Produit d'Hadamard des matrices.
$\nabla(J(\cdot))$	Gradient de la fonction J .
$diag(\mathbf{X})$	Éléments de la diagonale de la matrice \mathbf{X} .

Introduction

La plus ancienne carte géographique connue remonte à 2600 av. J.-C. en Mésopotamie, c'est depuis ce temps-là que les êtres humains ont commencé à s'intéresser à la localisation. Dans un premier temps, l'intérêt consistait à pouvoir s'orienter dans l'espace pour voyager et la localisation se faisait d'une manière approximative en se positionnant à proximité d'un lieu connu ou à l'intersection des lieux connus. Les évolutions qu'a connues l'humanité ont permis de développer les moyens de localisation et d'orientation en passant par l'utilisation des étoiles et des boussoles. Les avancées technologiques apparues ensuite ont donné un nouveau souffle à cette discipline, ce qui a révolutionné l'accès à l'information de la position. En effet, ceci a donné naissance naturellement à l'idée de fournir des services en fonction de la position de l'utilisateur. Plus récemment et avec l'avènement de l'Internet des Objets, l'information sur la position a suscité davantage d'intérêt afin de permettre une utilisation optimale des mesures de l'environnement remontées par les différents objets.

L'Internet des objets (IoT) est un système où un réseau d'objets intelligents interconnectés échangent des données sans intervention humaine [1–3]. Le développement d'applications dans l'IoT est fortement lié à l'information de localisation. Par conséquent, les technologies de localisation auront un rôle important dans l'IoT et pourraient être intégrées dans l'infrastructure ou dans l'objet lui-même. En fait, les données collectées sont rapportées pour une application IoT spécifique, ce qui nécessite des outils d'analyse de données dédiés pour les analyser et prendre les mesures appropriées. De nombreuses applications sont liées aux services de localisation, ce qui accroît l'importance des informations de localisation. Ces informations peuvent être utilisées pour le suivi de cibles, les applications de surveillance, le guidage de véhicules autonomes, etc [4–7]. Par conséquent, les données collectées sont inutiles, si elles ne sont pas combinées avec l'emplacement précis du noeud de capteur concerné. Cet emplacement peut être obtenu à l'aide de systèmes de navigation globale par satellite (GNSS), tels que le système de positionnement global (GPS) [8], qui est un système de localisation efficace en extérieur. Ces solutions ne peuvent pas être déployées dans des environnements intérieurs en raison des effets de trajets multiples causés par les obstacles existants entre les satellites et les utilisateurs et des pertes lors des traversées des murs et plafonds. Ce qui entraîne une dégradation importante des signaux GPS. Pour surmonter cette limitation, l'idée est d'utiliser des signaux radio pour la communication entre objets. La technologie de communication est également un défi dans le développement de l'IoT, et le choix est étroitement lié à l'application. Bluetooth [9], l'UWB (Ultra-Wide Band) [10], l'identification par radiofréquence (RFID) [11] et le réseau local sans fil (WiFi) [12] ont été largement utilisés pour la localisation en intérieur. La plupart des systèmes de localisation en intérieur proposés reposent sur des signaux WiFi en raison de la large utilisation des

appareils mobiles intégrant cette technologie. En fait, les autres solutions de communication mentionnées nécessitent une infrastructure spécialisée (balises radio sans fil) à intégrer dans l'environnement intérieur et l'ajout des équipements supplémentaires. Les caractéristiques du signal étant fortement liées à la distance entre l'émetteur et le récepteur, elles peuvent être utilisées pour effectuer la localisation [13–15]. Les paramètres faciles à obtenir sont généralement les suivants : indicateur de puissance du signal reçu (RSSI) [16–18], information sur l'état du canal (CSI) [19,20], angle d'arrivée (AOA) [14,21], temps d'arrivée (TOA) [22] et différence du temps d'arrivée [23]. Le RSSI ne nécessite pas de matériel spécifique pour la synchronisation du temps ou de la phase et aucune modification n'est nécessaire sur le logiciel du périphérique pour pouvoir l'acquérir. Ce qui fait de lui le paramètre le plus exploré en localisation en intérieur aujourd'hui.

Les méthodes existantes à base de RSSI peuvent être essentiellement classées en deux catégories : des solutions basées sur les fingerprints et des solutions basées sur la trilatération. La trilatération utilise des propriétés géométriques pour estimer l'emplacement du capteur [24]. La distance à un Point de Référence (PR) est estimée à l'aide d'un modèle de propagation et au moins quatre RPs sont nécessaires pour obtenir une position 3D. Cela nécessite que les RPs aient des positions connues, ce qui n'est pas facile à obtenir dans des conditions de propagation en intérieur (existence des obstacles, limitation de la portée radio, etc), qui est l'inconvénient majeur de la technique de trilatération. Pour améliorer les performances de localisation enregistrée par cette technique, nous proposons d'estimer toutes les distances par paire entre les capteurs et obtenir une matrice de distances, appelée 'Matrice de Distances Euclidiennes (MDE)' [25], entièrement connue. Par conséquent, le développement des solutions, qui permettent de récupérer une MDE complète à partir d'une matrice incomplète à entrées partiellement connues, pour fournir une information de distance fiable à chaque noeud capteur semble être intéressant. La méthode de fingerprinting surmonte les inconvénients mentionnés rencontrés par la méthode de trilatération car elle utilise une carte radio construite appelée 'radio map' à comparer aux mesures RSSI associées au capteur à localiser [26]. Cette comparaison peut être assurée d'une façon déterministe ou bien probabiliste. Il s'agit d'une solution efficace dont l'exactitude est liée à la suffisance des données, pour cela des bases de données RSSI volumineuses sont construites et manipulées pour obtenir une bonne précision de localisation. Cela accroît la complexité et la durée de fonctionnement des systèmes de localisation par fingerprints, les rendant ainsi inadaptés à la localisation en temps réel et incapables de gérer de grands réseaux de capteurs. Cette complexité peut être minimisée en développant un procédé d'optimisation qui permet d'estimer une matrice reliant la matrice de mesures RSSI aux coordonnées d'un noeud donné. Ce qui fait, nous développons un système analytique qui relie les puissances reçues des PAS au niveau d'un point et les coordonnées de ce point. Récemment, le développement des solutions qui utilisent des modèles d'apprentissage pour minimiser la complexité en ligne est fortement demandé. Cela est assuré par l'implémentation des solutions de localisation qui combinent la méthode de fingerprinting avec des méthodes de machine learning (ML). En effet, la préparation des données et le prétraitement est assuré dans la phase hors ligne / phase d'apprentissage, seule la tâche de prédiction est effectuée en ligne. Par conséquent, pour trouver sa position, un noeud de capteur interroge un modèle entraîné, qui estime la position. De tels algorithmes décalent la complexité de calcul du processus de prédiction en ligne vers la phase hors ligne. Ainsi, de telles solutions basées sur ML sont fortement

recommandées pour les applications de localisation en temps réel. Cela est motivé par l'efficacité des algorithmes de deep learning (DL) qui ont montré de très bonnes performances en localisation estimant, soit les coordonnées de localisation, soit d'autres informations de localisation.

L'objectif principal de cette thèse est d'améliorer les solutions de localisation existantes afin de développer des méthodes avancées fiables et précises tout en respectant les contraintes liées aux réseaux de capteurs (complexité de calcul et mémoire de stockage). Pour cela, nous avons contribué sur 2 niveaux.

- D'abord, nous avons essayé d'améliorer les performances de la trilatération en la combinant avec la technique de complétion de matrice en mettant en place un système de localisation distribué.
- Puis, des systèmes de localisation centralisés ont été développés en utilisant la technique de fingerprinting. Des méthodes probabilistes et déterministes ont été exploitées pour évaluer la similarité. Pour minimiser le temps et la complexité de la prédiction de position, l'utilisation d'un système, prédéfini en offline, reliant les puissances aux coordonnées semble être une bonne solution. Ce système peut être un système analytique ou un système d'apprentissage.

Le plan de ce manuscrit de thèse est précisé ci-dessous.

- **Le chapitre 1** est consacré à la présentation de l'état de l'art lié aux différents concepts utilisés tout au long de cette thèse. Au début, nous commençons par rappeler les caractéristiques de l'IoT ainsi que l'importance de la localisation pour assurer le bon fonctionnement de plusieurs applications. Après, nous détaillons les technologies et les principales techniques de localisation existantes, ainsi que les paramètres du signal radio exploités.
- **Dans le chapitre 2**, nous visons à améliorer les performances de localisation enregistrées par la technique de trilatération en estimant toutes les distances entre capteurs en déployant la méthode de complétion de matrice. Pour cela, nous avons utilisé des méthodes d'optimisation avancées.
- **Le chapitre 3** propose différentes approches de localisation pour une architecture centralisée basée sur le fingerprinting. Il commence par une étude comparative entre différentes méthodes d'évaluation de similarité et des méthodes d'estimation de position, afin de trouver la meilleure combinaison. Cette étude comparative est suivie par un processus de sélection de points d'accès mis en place afin d'éliminer l'information redondante pour ne garder que l'information utile. Pour alléger la complexité de calcul et de prédiction en ligne, un processus d'optimisation est développé permettant d'estimer une matrice reliant les puissances reçues au niveau d'un objet à ses coordonnées. Vu les contraintes de calcul liées à la mise en place d'un tel système analytique, un framework de localisation avancé combinant la méthode de fingerprinting et les réseaux de neurones convolutifs a été proposé. En s'inspirant des excellentes performances du CNN dans la classification d'images, nous avons transformé le problème de localisation en un problème de classification qui permet de prédire la classe contenant l'image radio correspondante au capteur à localiser.
- **La conclusion** résume les contributions de la thèse, ainsi que les principaux résultats

obtenus et on met en évidence les perspectives et les directions de recherche futures possibles.

Liste des publications

Journaux

1. W. Njima, I. Ahriz, R. Zayani, M. Terre et R. Bouallegue "Deep CNN for Indoor Localization in IoT-Sensor Systems", *Sensors*, vol. 19(14), p. 3127, 2019.
2. W. Njima, R. Zayani, I. Ahriz, M. Terre et R. Bouallegue "Beyond Stochastic Gradient Descent for Matrix Completion Based Indoor Localization", *Applied Sciences*, vol. 9(12), p. 2414, 2019.

Conférences

1. W. Njima, M. Terre, I. Ahriz, R. Zayani et R. Bouallegue "Localization by Inversion of the Taylor Expansion of the Received Power", PIMRC, Septembre 2019, Istanbul, Turquie.
2. W. Njima, I. Ahriz, R. Zayani, M. Terre et R. Bouallegue "Smart probabilistic approach with RSSI fingerprinting for indoor localization", SOFTCOM, Septembre 2017, pp.1-6, Split, Croatie.
3. W. Njima, I. Ahriz, R. Zayani, M. Terre et R. Bouallegue "Comparison of similarity approaches for indoor localization", WiMob, Octobre 2017, pp. 349-354, Rome, Italie
4. W. Njima, I. Ahriz, R. Zayani, M. Terre et R. Bouallegue "Extended Abstract : Error Density for Indoor Localization Based on RSSI fingerprint", ISWCS, Septembre 2017, Bologne, Italie.

Chapitre 1

La localisation indoor : Objectifs, applications et techniques

1.1 Introduction

Le besoin de se localiser ou de localiser un objet communiquant ou un utilisateur est devenu primordial avec l'émergence de l'Internet des Objets (IoT). En effet, les applications liées à l'IoT concernent plusieurs domaines (sécurité et surveillance, maisons intelligentes, besoins commerciaux et marketing, etc) et l'information de localisation conditionne le bon fonctionnement de ces applications. La localisation est essentiellement problématique dans les milieux intérieurs à cause de l'absence des signaux GPS. Le GPS s'est imposé en extérieur où la qualité du signal reçu et les conditions de propagation permettent d'atteindre des précisions de localisation acceptables. Cependant, il n'est pas adapté aux environnements intérieurs où la contrainte de visibilité directe entre l'utilisateur et les satellites n'est pas respectée. La localisation dans les environnements intérieurs s'est alors basée sur l'utilisation des signaux de communication entre objets ou entre les objets et une entité centrale. On distingue alors deux catégories. Dans la première, la tâche de localisation est assurée par une entité centrale (un serveur) qui assure la prédiction des positions en utilisant les informations collectées en masse par les différents objets. Dans la deuxième, le calcul de la position va être distribué et se fera au niveau de chaque objet.

D'autre part, selon la technique utilisée pour l'estimation de la position, on distingue deux catégories d'algorithmes.

- La range based, dans laquelle une première étape consiste à estimer les distances qui séparent l'objet à localiser de plusieurs positions de références ; puis l'estimation de la position se fait dans une seconde étape.
- La range free, dans laquelle la position d'un objet est directement estimée en utilisant les paramètres des signaux échangés.

Les différentes techniques de localisation seront présentées plus en détails dans ce chapitre après une introduction à l'IoT, la présentation des technologies de communication et des paramètres pouvant fournir une information sur la position.

1.2 L'Internet des Objets (IoT)

L'Internet des Objets (IoT) est considéré comme étant une nouvelle évolution de l'Internet, où chaque objet connecté possède sa propre identité, des capacités de calcul et des capacités de communication de plus en plus sophistiquées, tout en ayant des ressources limitées, surtout en termes de puissance. Ce concept a été défini en 2012 par l'Union Internationale de Télécommunications (UIT) comme une " Infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ". L'IoT tourne autour de trois grands volets :

- Les objets physiques.
- Les réseaux de communication électroniques assurant la connectivité entre les objets.
- Le réseau qui assure le traitement des données issues des objets.

Les données extraites des objets physiques sont analysées afin de pouvoir interagir et coopérer les uns avec les autres. Ces données permettent d'informer les utilisateurs (individu ou entreprise) en temps réel de l'évolution de leur environnement. Ce qui leur offre la possibilité d'agir en temps réel sur l'environnement concerné. Il existe des contextes qui permettent d'échanger l'information seulement entre les objets communicants et des entités spécifiques relevant d'une architecture centralisée (Figure 1.1). Par contre, la prise de décision peut se faire au niveau d'un serveur dédié ou bien au niveau des objets. Pour l'architecture distribuée, la communication entre objets est aussi possible (Figure 1.1). La prise de décision se fait au niveau d'un serveur dédié ou bien au niveau de l'objet. Le déroulement du processus de prise de décision dépend du service dédié par ce système.

Les possibilités d'applications et de services prometteurs qu'offrent ce contexte sont nombreuses et presque illimitées visant à améliorer les services de sécurité, santé, transport, éducation, protection des énergies, etc. De nombreux services et applications de l'IoT sont commercialisés sur le marché ou open source. En effet, des initiatives ayant pour but d'améliorer la qualité de vie, sont lancées, partout dans le monde, sous forme de "Villes intelligentes ou Smart cities". Au deuxième semestre de 2015, Apple a vendu 4.2 millions de montres connectées. Ce marché aura un grand potentiel et un fort impact d'ici 2020 comme le révèle une étude récente de l'Economist Intelligence Unit qui indique que ce secteur pèsera 4 milliards de dollars en 2024. Mais, des problèmes de maintenance et de sécurité restent posés. En plus, pour plusieurs applications mentionnées ci-après, les données échangées deviennent inutiles si elles ne sont pas combinées avec l'information précise de la position de l'objet concerné qu'il soit fixe ou mobile. C'est dans cette problématique que s'inscrit ce travail de thèse.

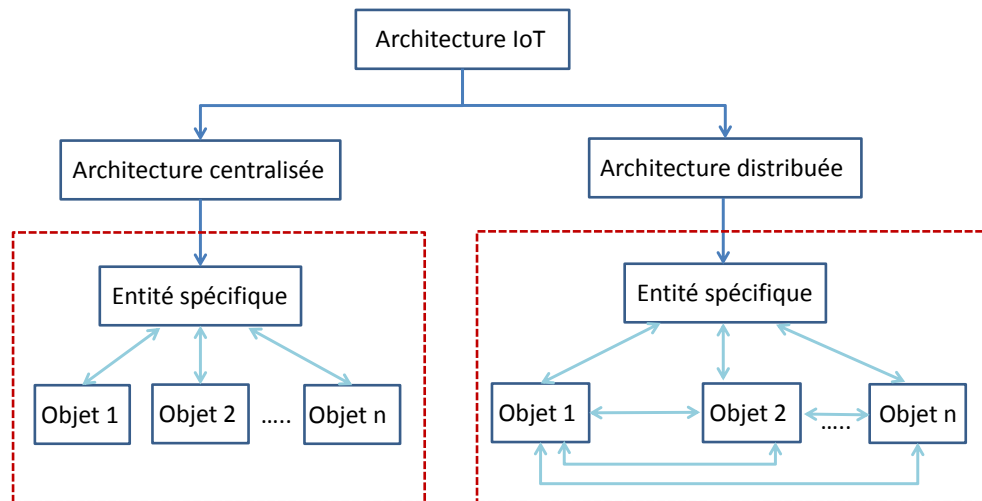


FIGURE 1.1: Les types d'architecture en IoT.

1.3 Applications diverses de l'IoT liées à la localisation

Une analyse de plus de 150 cas d'utilisations de l'IoT a été menée récemment par "McKinsey Global Institute" allant du domaine de la santé, afin d'assurer le bien être des utilisateurs à l'industrie pour garantir la sécurité des travailleurs et optimiser la maintenance. Cette analyse qui s'intéresse aux applications IoT prévoit que ce marché passera de 3.9 trillions de dollars à 11.1 trillions de dollars par an d'ici 2025.

Plusieurs applications exigent que les informations de localisation soient acquises rapidement et fiables afin qu'elles soient utilisées localement ou bien envoyées vers un autre objet. Il est important de fixer nos besoins en positionnement : position absolue (coordonnées géographiques) ou bien position relative (position par rapport à ce qui existe dans l'environnement). Selon la référence de la norme 3GPP TR 22.862 V14.1.0 Technical Specification Group Services and System Aspects [27], "Feasibility Study on New Services and Markets Technology Enablers for Critical Communications, Stage 1 (Release 14)," Sept. 2016. , dans 95% des cas y compris en intérieur, il faudra une précision de localisation inférieure à 1 mètre. Dans ce qui suit, on cite quelques applications qui nécessitent une information précise de position dans différents domaines (santé, sécurité, marketing, education, etc).

- Dans des situations d'urgence où les secours ne peuvent pas arriver rapidement sur les lieux, un véhicule aérien sans pilote (UAV), avec des capacités de localisation importantes, peut être utilisé pour accélérer l'intervention. Dans ce cas, les drones sont utilisés pour collecter des informations vidéos sur le site et livrer des équipements de secours. Ils renvoient également ses informations de position et autres données du site exact où l'accident s'est produit. Ces véhicules peuvent également être utilisés pour rechercher un intrus ou un suspect et surveiller les catastrophes naturelles.
- Plusieurs services dédiés aux villes intelligentes, connues sous le nom de smart cities, sont fournis y compris l'éclairage et la commande à distance d'appareils électroména-

gers. Le système reconnaît le nombre d'utilisateurs présents dans une pièce. Il vérifie régulièrement la position des noeuds pour déterminer le nombre des noeuds dans une pièce et lorsqu'il n'y trouve aucun, il éteint la lumière.

- La notion de Smart Shopping est apparue avec l'IoT comme la diffusion des informations commerciales contextuelles ou personnalisées telles que des messages publicitaires dépendants de la position et dirigés vers un groupe physique (publicité géo-ciblée). Cela est assuré en récupérant progressivement le comportement des utilisateurs mobiles qui font l'objet d'un suivi dans un centre commercial.
- Une application typique, où une précision importante de localisation est requise, est la prévention de la collision des véhicules : chaque véhicule doit être conscient de sa propre position, de la position des véhicules à proximité, ainsi que de leurs chemins prévus, afin d'éviter les collisions. Pour cela, l'échange des positions se fait chaque 100 ms avec une précision inférieure ou égale à 1 m.
- Différents services de Smart health sont introduits parmi lesquels on cite le service de surveillance de personnes à risque tel que celui qui offre une aide personnelle et un suivi aux personnes diabétiques. Il consiste en un dispositif permettant de suivre les mouvements des personnes tous les jours. Il peut être connecté à un analyseur de sang pour enregistrer en temps réel les paramètres vitaux du sang et il peut enregistrer les principales activités de la journée (balades, plan de repas, etc.) pour fournir une aide si nécessaire et pour faciliter la communication avec les spécialistes de la santé (médecins, etc.).
- Smart transportation ou le transport intelligent comme par exemple, pour offrir une aide aux autobus scolaires et au transport des enfants. Le conducteur sait combien d'enfants, identifiés par des étiquettes, sont supposés être à bord lorsque l'autobus est prêt à partir. Un système de localisation intérieur très précis est donc capable de localiser exactement toutes les étiquettes à l'intérieur de l'autobus et permet aux portes d'être fermées uniquement lorsque tous les enfants sont en sécurité. Il est clair qu'un tel système doit être très précis : il est obligatoire d'éviter qu'un enfant s'approchant de l'autobus, ou pire encore, prenant l'escalier d'entrée, ne soit bloqué par les portes à la suite d'une erreur de localisation de quelques centimètres.
- Applications multiples de guidage de personnes : On peut citer le guidage adapté à des personnes avec des déficiences visuelles dans des milieux urbains et dans des milieux intérieurs où l'aménagement actuel ne tient que rarement en compte de ce type de handicap. Le guidage des pompiers intervenant dans des milieux à risques est bien évidemment aussi important afin de les empêcher de s'engouffrer dans des voies dangereuses lors des interventions. Il y a aussi, la mise en oeuvre des visites touristiques guidées, pour tout public, comme le guidage d'une visite dans un musée qui propose à chaque personne un parcours personnalisé par le biais d'un système individuel portatif.
- Un déficit en terme de localisation est envisageable avec l'apparition de la 6ème génération, censée être commercialisée en 2030. En effet, selon 6G KPI réalisé par HUAWEI, 6G exige 1 cm de précision en intérieur / 0.1 ms et 50 cm de précision en extérieur / 0.1 ms afin de promouvoir des nouveaux services et applications et envisager d'autres applications.

1.4 Les technologies de communication utilisées pour des fins de localisation en intérieur

Quelque soit l'application visée, il y a une nécessité d'échanger des informations. Plusieurs technologies peuvent être envisagées selon les contraintes de débit, portée, consommation d'énergie et environnement de propagation. En plus de véhiculer des informations liées à l'application, ces technologies peuvent être utilisées pour estimer la position. Comme précisé au début de ce chapitre, la localisation en extérieur est assurée par le GPS qui garantit un niveau de localisation satisfaisant. Les signaux GPS en intérieur subissent une dégradation et même un blocage suite aux obstacles existants entre les utilisateurs et les satellites. Dans cette section, plusieurs technologies alternatives existantes qui ont été utilisées pour fournir des services de localisation en intérieur seront présentées et discutées. Plusieurs technologies émergentes basées sur la lumière visible, l'acoustique, le champs magnétique et les techniques d'imagerie seront abordées en premier, puis on se concentrera sur les technologies de communication radio.

1.4.1 Technologies émergentes exploitées dans la localisation

Une solution est d'utiliser un organe dédié au niveau de l'objet pour faire de la localisation. Dans ce cadre plusieurs technologies émergentes ont vu le jour que nous allons présenter dans cette section.

Champs magnétique

Une solution pour assurer une localisation en intérieur est d'utiliser des mesures des champs magnétiques afin de générer la carte des lieux où on se trouve [28]. Cette technologie est associée à une précision importante < 1 m. Ces systèmes ont essentiellement été exploités dans les entrepôts pour le guidage d'objets où de robots, la capture de mouvements et la réalité augmentée [29]. Cette technologie est peu répandue en localisation et très peu utilisée dans les lieux publics (gares, aéroports, musées, etc) vu la limitation de la cartographie disponible, pourtant, diverses utilisations possibles lui sont associées.

Lumière visible

La communication par lumière visible (VLC) est une technologie émergente pour le transfert de données à grande vitesse en utilisant la lumière visible entre 400 et 800 THz, émise principalement par des diodes LED [30]. Les techniques de localisation basées sur la lumière visible utilisent des capteurs de lumière pour mesurer la position et la direction des émetteurs à LED. Ces émetteurs transmettent le signal qui peut être utilisé pour la localisation. Ce qui fait de la lumière visible un bon candidat pour la localisation en intérieur est la possibilité d'exploiter les infrastructures existantes pour l'illumination en localisation [31]. En plus, une telle solution peut être introduite dans les environnements où les fréquences radio ne sont pas autorisées, comme les hôpitaux, les mines, les usines de pétrochimie, etc. Mais, une limite fondamentale est la nécessité de visibilité directe entre la LED et le(s) capteur(s) pour assurer une localisation précise, ce qui entraîne un coût élevé de mise en place.

Signal acoustique

La localisation basée sur le signal acoustique exploite les microphones omniprésents dans les smartphones pour capturer les signaux acoustiques émis par des sources sonores et estimer l'emplacement des utilisateurs par rapport à ces sources [32]. Plusieurs systèmes de localisation en intérieur qui utilisent les ondes sonores ont été développés [33,34] en utilisant des capteurs acoustiques placés dans des positions fixes. Dans [35], le système proposé définit des signatures à chaque espace de l'environnement intérieur étudié selon les différentes sources de bruit qui puissent le distinguer (ordinateurs, climatiseurs, etc). Cette signature permet d'identifier et caractériser un espace. Les systèmes de localisation à base acoustique atteignent une précision de localisation élevée. A cause des limitations du microphone des smartphones (fréquence d'échantillonnage / filtre anti-aliasing), seuls les signaux acoustiques dans la bande audible (< 20 kHz) peuvent fournir des estimations de localisation précises. Le besoin d'infrastructure supplémentaire (c.-à-d. sources acoustiques / noeuds de référence) et le taux de mise à jour élevé (qui impacte la batterie de l'appareil), font que le signal acoustique n'est pas une technologie très répandue pour la localisation.

Ultrasons

La technologie de localisation par ultrasons (onde de nature mécanique et élastique, qui est diffusée par un objet ou un corps [36]) repose principalement sur les mesures du temps d'arrivée de signaux ultrasonores (> 20 kHz) et la vitesse du son pour calculer la distance entre un émetteur et un noeud récepteur [37, 38]. La plupart des systèmes de localisation à base d'ondes ultrasons, combinent ces ondes avec une autre information afin d'estimer la distance entre l'émetteur et le récepteur pour prédire la position occupée par l'objet à localiser. Les informations provenant des ultrasons sont, par exemple, combinées avec celles provenant d'une interface RF dans le système proposé dans [39]. Il a été prouvé qu'un système de localisation par ultrason offrait une précision de localisation en centimètres avec une possibilité de suivre plusieurs noeuds mobiles en même temps avec une efficacité énergétique élevée. Habituellement, la transmission du signal ultrasonore est accompagnée d'une impulsion RF pour fournir la synchronisation nécessaire. Cependant, contrairement aux signaux RF, la vitesse du son varie considérablement lorsque l'humidité et la température changent. Enfin, même si des algorithmes complexes de traitement du signal peuvent filtrer les niveaux élevés de bruit ambiant pouvant dégrader la précision de la localisation, une source de bruit permanente peut continuer de dégrader gravement les performances du système.

Les techniques d'imagerie

Cette méthode consiste à équiper le milieu intérieur avec un ensemble de caméras placées à des positions connues. Ces capteurs analysent l'environnement et détectent les anomalies ainsi que les intrus en repérant un objet et en déduisant sa position. Pour cela, différents types de caméras peuvent être déployés (caméra simple, caméra dotée de l'infrarouge, caméra thermique, etc). Ces techniques d'imagerie sont fréquemment utilisées pour la surveillance et la sécurité des lieux et elles sont très répandues en robotique [40–42]. Les robots autonomes se déplacent dans l'environnement grâce aux systèmes de vision comme le robot Pearl Guard qui est un robot mobile de surveillance et d'alerte à distance [43]. Il est équipé de 4 caméras

infrarouges placées de façon à ce qu'il couvre tout l'environnement et une caméra thermique. Ce robot renvoie une alerte dès qu'il détecte une intrusion en renvoyant la position de l'intrus estimée en combinant l'information provenant des caméras et des autres capteurs intégrés dans le robot. Les systèmes de localisation à base des techniques d'imagerie garantissent souvent un niveau de précision important en milieu intérieur. Cependant, des capacités de calcul et de vision élevées sont requises, ce qui augmente considérablement le coût de déploiement d'une telle solution de localisation.

L'utilisation des capteurs inertiels

Ce sont des techniques de localisation sans infrastructure, qui exploitent les capteurs inertiels [44, 45] (accéléromètres, odiomètre, gyroscopes, etc.) intégrés dans une carte embarquée, permettant d'estimer la position actuelle à partir de la dernière position connue. Pour cela, il est nécessaire de renseigner une position d'origine pour ce type de système qui présente l'avantage d'être totalement autonome. Cependant, des erreurs de localisation peuvent s'accumuler progressivement engendrant une dégradation importante de la précision de localisation. La localisation à base des capteurs inertiels est souvent utilisée dans l'aéronautique et la navigation marine. L'estimation de positions des objets est effectuée à des intervalles réguliers et l'erreur peut grandir considérablement. Ces capteurs sont, de préférence, utilisés comme support pour d'autres technologies, afin d'affiner la localisation selon un principe de fusion de données.

1.4.2 Les technologies de communication radio

Dans cette section, on présentera brièvement les technologies de communication radio utilisées dans le contexte de localisation. Parmi ces technologies, on peut citer WiFi, Bluetooth, Zigbee, RFID, Ultra large bande (UWB), etc. Ces solutions de localisation présentent l'avantage d'être à moindre coût car tous les objets connectés possèdent une technologie de communication.

Bluetooth (BLE) ou le standard 802.15.1

Bluetooth utilise les bandes ISM 2.4 GHz et 5 GHz, largement utilisé pour les communications à courte distance 10 – 100 m [46]. Il utilise une puissance de transmission très faible engendrant une couverture plus courte que les autres technologies WLAN. Par conséquent, Bluetooth ne convient pas à la localisation grande surface qui nécessite une infrastructure importante en mettant beaucoup de balises Bluetooth dans l'environnement. Cependant, plusieurs entreprises se sont lancées sur des solutions de localisation basées sur le Bluetooth Low Energy comme les start-ups françaises Insiteo [47] et Polestar [48]. Ces entreprises multiplient le nombre des balises BLE et proposent de profiter de sa portée limitée afin d'augmenter la précision. Evidemment, le problème majeur est la nécessité d'installer un nombre important de balises dans l'environnement où on veut se localiser, surtout lorsqu'on vise les grands espaces (aéroports, musées, etc).

Zigbee ou le standard 802.15.4

Zigbee utilise une bande autour de 868 MHz en Europe, 915 MHz aux Etats-Unis et en Australie et 2.4 GHz dans d'autres régions [49]. ZigBee est utilisé pour la transmission entre appareils dans un réseau maillé sans fil, avec une portée de 100 m, à un faible coût, faible débit, court temps de latence et faible consommation d'énergie. Plusieurs applications de localisation basées sur cette technologie ont été développées surtout celles qui exploitent la puissance reçue. Son intérêt principal est d'avoir un niveau important d'autonomie compte tenu du faible coût et de la gestion optimale d'énergie.

Radio Frequency Identification (RFID)

Cette technologie fait interagir deux parties : étiquette et lecteur [50]. Le lecteur utilise un champ électromagnétique radiofréquence pour lire les données de l'étiquette et identifier l'objet associé à l'étiquette. La balise ou l'étiquette peut avoir une batterie ou non, ce qui en fait une étiquette active ou une étiquette passive. L'étiquette passive RFID souffre des problèmes de collision de balises et de collision de lecteurs. La collision de balises survient lorsqu'un lecteur lit plusieurs étiquettes, et une collision entre lecteurs se produit lorsque la couverture de deux lecteurs se chevauche en détectant la même balise en même temps. La portée de la communication RFID est très courte. Cette technologie a donné naissance à une vaste gamme de systèmes de localisation [51, 52]. Deux principes de localisation sont possibles [52] :

- Localisation d'une étiquette RFID associée à l'objet d'intérêt par un réseau de capteurs.
- Localisation du lecteur des étiquettes RFID qui sera attaché à l'objet à localiser. Ce dernier déterminera sa position en s'aidant des étiquettes placées dans des positions connues dans l'environnement approprié.

Ultra Wideband (UWB)

En mode UWB, des impulsions ultra courtes d'une durée < 1 ns sont transmises sur une large bande passante (> 500 MHz) [53]. Sa transmission peut être considérée comme un bruit de fond par rapport aux autres technologies sans fil, donc, en théorie, il peut utiliser n'importe quel spectre sans interférer avec d'autres utilisateurs. Il utilise une faible puissance de transmission -41.4 dBm / MHz, ce qui signifie que sa consommation d'énergie est faible. Un autre avantage de UWB est son immunité aux chemins multiples. Ce qui fait de UWB un candidat particulièrement intéressant pour la localisation en intérieur qui peut garantir des performances prometteuses. Les techniques de localisation associées à cette technologie se basent généralement sur les paramètres temporels qui seront présentés au cours de ce chapitre. S'il est possible d'obtenir précisément l'instant d'arrivée du trajet direct entre l'émetteur et le récepteur, l'application d'un algorithme approprié permet d'estimer la position de l'objet cible. Cependant, la lenteur des progrès dans l'élaboration de la norme UWB a limité l'utilisation et l'intégration de la technologie UWB dans les dispositifs utilisateurs portables.

Réseaux mobiles (2G, 3G, 4G)

Ces trois technologies sont basés sur les réseaux cellulaires qui fonctionnent sur des bandes de fréquences réservées, cela les rend moins sensibles aux interférences. Le réseau 4G couvre déjà la plupart des bâtiments, il n'y a donc pas besoin d'une infrastructure supplémentaire. Des études ont montré que des revenus énormes peuvent être générés par l'exploitation du réseau 4G pour la localisation [54–56] et de plus en plus les opérateurs téléphoniques s'y sont intéressés. Plusieurs techniques de localisation existent : des méthodes d'identification de la cellule où se trouve le mobile (Cell ID), des méthodes basées sur des mesures temporelles et des méthodes basées sur les directions d'arrivée. La méthode E-OTD (Enhanced - Observed Time Difference) a été développée par Cambridge Position Systems. Le téléphone mobile envoie un signal aux stations de base à proximité. La réponse de la station la plus proche sera analysée par un serveur dédié pour localiser le mobile dans la zone de couverture de la station de base. Le temps requis pour localiser un objet mobile est d'environ 5 secondes avec une précision de 30 – 50 mètres [57].

WiFi ou le standard 802.11

C'est l'une des technologies sans fil les plus utilisées qui utilise deux bandes sans licence aux fréquences 2.4 GHz et 5 GHz [58, 59]. La plupart des bâtiments (centres commerciaux, bureaux, etc) ont déjà déployé des points d'accès WiFi qui couvrent l'ensemble des bâtiments. Ce qui fait que les systèmes de localisation utilisant le WiFi n'auront pas besoin d'infrastructures supplémentaires pour être mis en place engendrant un coût très faible de mise en place. En plus, la plupart des produits commerciaux (les smartphones, les ordinateurs portables et les tablettes) supportent le WiFi. Ce qui fait du WiFi un candidat idéal pour la localisation en intérieur largement étudié et déployé. On peut citer, par exemple, le projet [60, 61] mis en place par Intel Research Seattle, Cambridge et les Universités de Washington et de San Diego pour améliorer les performances de méthodes existantes de localisation. Pour cela, un mobile avec une interface GPS et une interface WiFi se localise en extérieur avec le GPS sinon avec le WiFi qui couvre cette zone.

1.5 Les paramètres du signal radio

Plusieurs paramètres de signal radio peuvent être utilisés et exploités dans la localisation dans le cas d'exploration des technologies de communication radio. On détaillera ci-dessous les plus utilisés.

1.5.1 L'angle d'arrivée (AoA)

Cette approche repose sur l'exploitation des angles d'arrivée du signal émis par l'objet d'intérêt au niveau des positions de référence connues [62] qui sont les points de mesure. Cette technique est illustrée dans la Figure 1.2.

La zone possible contenant le mobile est donnée par l'intersection des faisceaux passant par chaque récepteur. Lorsque le mobile se trouve sur la même droite qui relie les deux récepteurs, l'estimation de position n'est plus possible. D'où la nécessité d'avoir un récepteur supplémentaire. La mesure de ce paramètre nécessite des antennes directives ou des réseaux

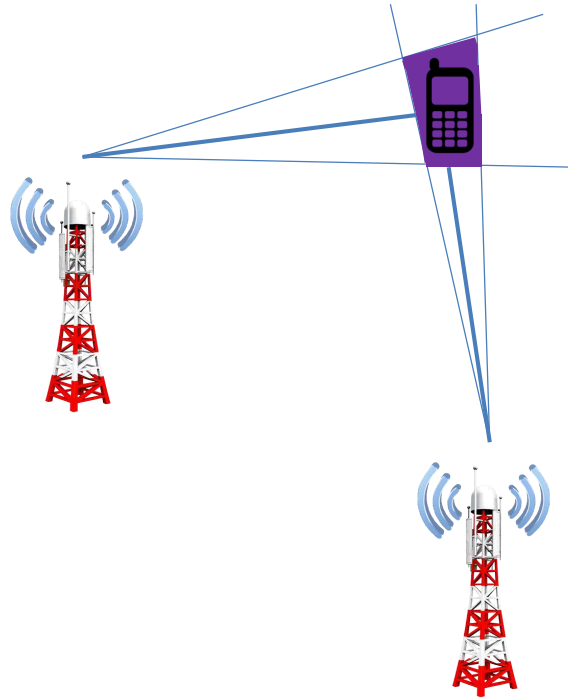


FIGURE 1.2: Principe de l'AoA dans la localisation.

d'antennes et un étalonnage minutieux. Tandis que sa précision se dégrade lorsque la distance émetteur-récepteur augmente. De plus, en raison des effets de trajets multiples dans les environnements intérieurs, l'AoA est souvent difficile à obtenir pour la non visibilité directe entre l'émetteur et le récepteur [63]. Parmi les travaux qui exploitent ce paramètre, Sextant [64] propose à un utilisateur d'orienter son téléphone vers un PA WiFi afin d'utiliser le gyroscope du smartphone pour calculer l'angle d'arrivée.

1.5.2 Temps de propagation des signaux

C'est le temps entre l'instant de propagation du signal et l'instant de réception de ce signal. Lorsque l'émetteur et le récepteur sont parfaitement synchronisés, on parle du temps d'arrivée (ToA) [65] ou temps du vol (ToF). Cette mesure peut être utilisée pour calculer la distance, entre le mobile à localiser et les noeuds de référence, exploitée surtout par les méthodes de localisation géométriques. Pour éliminer les importantes erreurs de localisation, une contrainte de visibilité directe entre l'émetteur et le récepteur s'impose. Cela est dû au fait que les obstacles dévient les signaux émis qui traversent ensuite un chemin plus long provoquant une augmentation du temps pris pour que le signal se propage.

Et si la synchronisation entre l'émetteur et le récepteur n'est pas assurée, on parle de la différence du temps d'arrivée (TDoA) [66] qui exploite les temps de propagation des signaux. La précision de localisation assurée par ce paramètre dépend de l'existence d'une

visibilité directe entre l'émetteur et le récepteur et exige une synchronisation stricte entre les émetteurs.

Avec une synchronisation modérée entre l'émetteur et le récepteur, le Round-Trip Time of Flight (RToF) peut être utilisé. RToF mesure le temps de propagation du signal aller-retour (c'est-à-dire émetteur-récepteur-émetteur). Un problème important avec les systèmes RToF est le délai de réponse du récepteur qui dépend de l'électronique du récepteur et du protocole utilisé. Ce délai ne peut pas être ignoré pour les systèmes de localisation en intérieur qui nécessitent un temps de réponse rapide.

1.5.3 La réponse impulsionnelle du canal (CIR)

La réponse impulsionnelle du canal (CIR) est une information utilisée pour corriger l'effet du canal. Cette réponse impulsionnelle estimée peut être transmise aux couches supérieures en tant que Channel State Information (CSI). L'estimation de la réponse impulsionnelle capture à la fois les réponses en amplitude et en phase du canal. Ces mesures sont stables et garantissent une précision de localisation élevée. Cependant, un accès à la couche physique du réseau est nécessaire.

1.5.4 Phase d'arrivée (PoA)

Les approches basées sur la PoA utilisent la phase ou la différence de phase de signal de porteuse pour estimer la distance entre l'émetteur et le récepteur. Une hypothèse courante pour déterminer la phase du signal côté récepteur est que les signaux transmis des nœuds de référence sont de forme sinusoïdale pure ayant la même fréquence. Il existe un certain nombre de techniques disponibles pour estimer la plage de distances ou la distance entre l'émetteur et le récepteur en utilisant la PoA. La PoA peut être utilisée conjointement avec RSSI, ToF, TDoA pour améliorer la précision de localisation et améliorer les performances du système. Le problème avec l'approche basée sur la PoA est qu'elle nécessite une visibilité directe pour une précision importante, ce qui est rarement le cas en intérieur.

1.5.5 La puissance reçue du signal

À part l'angle, le temps et la réponse du canal radio, la puissance du signal reçu Received Signal Strength Indicator (RSSI) [67], par un équipement mobile d'un point de référence dont la position est connue, est une propriété très répandue. Cette mesure associée à un modèle de propagation, qui caractérise l'affaiblissement du signal tout au long du chemin, nous permet de déterminer la distance entre l'équipement à localiser et le point de référence concerné. Ce qui peut être très utile à exploiter par plusieurs algorithmes de localisation simples. La puissance de signal reçue nous renseigne sur la qualité de la communication quelque soit la technologie radio utilisée, ce qui rend facile l'adaptation des algorithmes de localisation à base de RSSI à tous types d'environnements. En plus, l'acquisition des RSSI est simple et ne nécessite pas des équipements supplémentaires dédiés, cela se fait avec des équipements de commerce largement utilisés. Ce qui représente un atout important et fait que ce paramètre soit le plus utilisé dans la littérature dans le contexte de la localisation. Cependant, cette valeur de puissance est variable dans le temps dû aux effets des multi-trajets et aux affaiblissements. Ce qui nécessite la mise en place d'algorithmes de localisation

plus sophistiqués et plus complexes. Dans le reste de nos travaux, nous allons exploiter ce paramètre.

1.6 Les techniques de localisation

Comme mentionné dans l'introduction, la localisation fait l'objet de plusieurs travaux de recherche. Nous citons dans cette partie les méthodes fondamentales de localisation liées à la localisation des capteurs. Ces méthodes peuvent être classées en deux catégories [68, 69] : les méthodes range based qui se basent sur une évaluation de la distance entre émetteur et récepteur et les méthodes range-free, qui sont considérées comme économiques en matériel puisqu'elles se basent sur l'information de connectivité. Dans notre travail, on se focalisera sur la première catégorie dont on détaillera les principales méthodes (des méthodes géométriques comme la trilatération et la triangulation, le fingerprinting).

1.6.1 Proximité

C'est la méthode de localisation la plus simple [70]. Si l'utilisateur est dans la zone de couverture d'une station de base de position connue, la position de cet utilisateur par rapport à la station connue peut être approximée. La plupart des systèmes de localisation basés sur la 4G utilisent cette méthode. Une précision entre 50 - 200 m, qui correspond à la taille d'une cellule en environnement urbain dense, est atteinte. C'est une méthode qui peut être utilisée quelque soit la technologie. Typiquement le bluetooth et le RFID se basent sur cette technique. La précision dépend de la portée de la technologie. Les méthodes basées sur la proximité ont une variance élevée qui peut parfois ne pas satisfaire le besoin de localisation en intérieur.

1.6.2 Les méthodes géométriques

- **Trilatération** [71, 72] : cette méthode peut être implémentée en utilisant la puissance du signal reçu et le temps de propagation du signal. Ces paramètres sont proportionnels à la distance entre l'émetteur et le récepteur : le temps d'arrivée est multiplié par la vitesse de propagation et un modèle de propagation est appliqué à la puissance du signal reçu. La position du capteur peut être estimée si on connaît les distances qui le séparent d'au moins trois positions connues.

Soient trois noeuds qui vont servir comme points de référence (A_1, A_2, A_3) dont on connaît les positions (x_1, y_1) , (x_2, y_2) et (x_3, y_3) . Soit un point M dont on vise estimer la position (x, y) . La distance qui sépare le point de référence i et le point M est calculée en utilisant un modèle de propagation. La relation entre le point M et les points de référence est exprimée dans l'équation 1.1 dont le principe est illustré dans la Figure 1.3. Cette méthode ainsi que les équations de résolution seront détaillées dans le chapitre suivant.

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2 \end{cases} \quad (1.1)$$

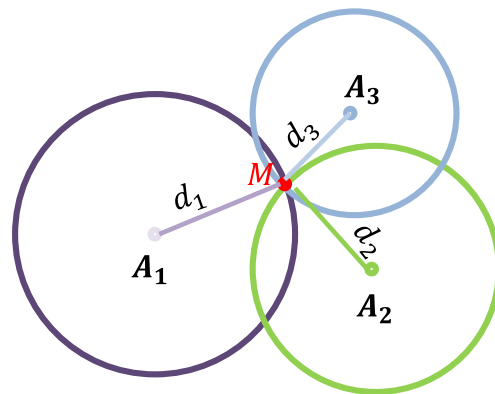
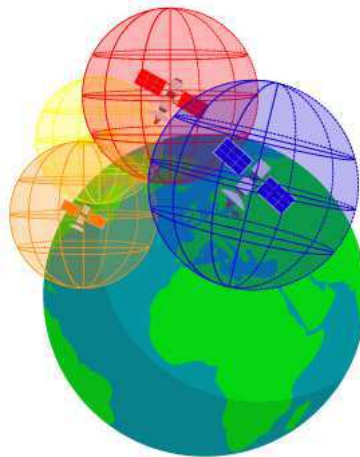


FIGURE 1.3: Principe de la trilatération.

Cette technique est utilisée par les systèmes GPS afin de localiser un récepteur GPS. En fait, les signaux transmis par les satellites dont les positions sont connues, qui présentent les points de référence, forment des sphères dont l'intersection correspond à la position estimée du récepteur (Figure 1.4).

FIGURE 1.4: Principe de fonctionnement du GPS (Source : <https://www.geotracteur.fr/>).

C'est une méthode simple à déployer et à implémenter. Cependant, elle est très sensible aux conditions de propagation qui sont complexes et difficiles à modéliser dans les conditions en intérieur. De plus, le nombre de points desquels le capteur reçoit le signal peut être limité à cause des conditions de propagation, ce qui peut dégrader

les performances en localisation. Dans un contexte IoT, plusieurs objets peuvent être localiser. Ainsi, les distances entre ces objets et un certain nombre de points de référence doivent être connues. Ces dernières peuvent être organisées dans une matrice de distance, appelée ‘matrice de distances euclidiennes (MDE)’ [25]. Chaque ligne et chaque colonne de la matrice correspond à un objet du réseau qui peut être un objet à localiser (avec une position inconnue) ou un point de référence (avec une position connue). A cause des conditions de propagation dans les milieux intérieurs, il se peut que la MDE ne soit pas complètement renseignée (si la puissance reçue par un objet est inférieure à sa sensibilité). Dans ce cas et pour fournir une information de distance fiable à chaque noeud capteur, la technique de complétion de matrice est ainsi proposée. Son rôle est de récupérer une MDE complète à partir d’une matrice incomplète à entrées partiellement connues. Plusieurs algorithmes et approches ont été proposés et étudiés pour approcher les distances manquantes dans les systèmes basés sur les mesures RSSI.

Les auteurs dans [73] formulent ce problème comme étant une minimisation des moindres carrés. Pour résoudre ce problème de minimisation, ils introduisent l’algorithme de Newton itératif modifié pour optimiser le critère [74]. Cette solution converge rapidement si les paramètres sont bien choisis mais ses performances sont très sensibles à l’initiation de ses paramètres. Elle peut servir comme une bonne estimation de localisation initiale à affiner par d’autres algorithmes de localisation plus avancés. Récemment, des approches basées sur Singular Value Thresholding (SVT) [75], Accelerated Approximal Gradient (APG) [76] et Augmented Lagrange Multiplier (ALM) [77] ont été proposées. La méthode SVT qui repose sur la décomposition en valeurs singulières (SVD) d’une matrice de rang faible, est associée à des mauvaises performances de complétion et à un taux de convergence faible. Des méthodes ALM et APG ont été proposées pour améliorer une initialisation SVT. Cependant, ces méthodes dépendent fortement du choix des paramètres et sont sensibles au bruit. Pour surmonter ce problème, un modèle de complétion matricielle régularisée est proposé dans [78] introduisant la multivariate function Bregman divergence pour approximer la MDE complète. L’inconvénient majeur d’une telle méthode centralisée est son coût de calcul. Les méthodes que nous avons mentionnées sont toutes basées sur les RSSI. Comme alternative aux mesures de RSSI, Nguyen et al. se concentrent sur les informations de connectivité / distance en mettant en place un protocole de routage [79]. Le problème de récupération de la matrice est formulé comme un problème d’optimisation sans contrainte dans Smooth Riemann Manifold. Ensuite, l’algorithme de gradient conjugué non linéaire est appliqué sur Smooth Riemann Manifold. Cette approche permet de récupérer la MDE dans des environnements avec et sans bruit. Cependant, cette approche se concentre uniquement sur les informations de position à deux dimensions. Les auteurs dans [80] classent le bruit pouvant être présent dans un environnement intérieur en trois catégories et permettent d’approximer les distances inconnues dans un tel milieu bruité. Cette approche assure une bonne précision de localisation par rapport aux autres systèmes existants. Cependant, c’est un système de localisation centralisé.

Une fois la matrice MDE remplie, la deuxième étape consiste à estimer l’emplacement des capteurs. Pour cela, les coordonnées peuvent être estimées à partir des distances en

appliquant un processus de factorisation. Cette factorisation exige que la matrice soit Semi Définie Positive (SDP) [81]. Cela peut être résolu efficacement et avec précision par le biais de la toolbox CVX [82] de Matlab. Cependant, si la matrice n'est pas SDP, le problème est résolu en introduisant des relaxations semi-définies (RSD) [83, 84]. La localisation basée sur RSD convertit le problème de localisation non convexe en un problème convexe et marque une bonne précision de localisation. Cependant, la complexité de calcul associée à ces techniques est étroitement liée à la taille du problème. Ce qui fait que cet algorithme est adapté aux réseaux de taille moyenne uniquement en raison de l'importance du temps d'exécution requis. Pour réduire ce temps d'exécution, une méthode de localisation à base de relaxation semi-définie pondérée est réalisée [85]. Elle vise à améliorer la précision de localisation, mais elle ne convient pas pour le traitement de réseaux de large dimension. Le problème de localisation de sources multiples est formulé dans [86] comme une unimodality constrained matrix factorization (UMF) et deux techniques de rotation ont été développées pour résoudre le problème. Dans [87], une eigendecomposition est d'abord appliquée pour trouver les emplacements locaux des noeuds, puis une matrice de rotation et un vecteur de translation sont utilisés pour les transformer en véritables positions. Zhang et al. dans [88] proposent d'estimer les positions des noeuds par l'algorithme classique de positionnement multidimensionnel (MDS) [89] utilisant une eigendecomposition tronquée. Cependant, la méthode MDS nécessite un taux de récupération élevé de la MDE, ce qui n'est pas garanti lorsqu'on travaille dans des environnements bruyants. Une fois la MDE complétée, l'estimation peut se faire en utilisant le fingerprinting, qui sera décrit dans la section suivante, ou la trilatération. Le principal inconvénient du fingerprint est l'utilisation d'une carte radio construite en offline. En outre, en raison des changements d'environnement récurrents, cette carte radio doit être fréquemment mise à jour. La méthode de la trilatération peut être appliquée si nous avons au moins trois distances connues séparant le noeud à localiser et les autres noeuds si la localisation est effectuée dans un espace 2D. Pour chaque noeud à localiser, la fusion de données est réalisée en combinant des mesures provenant de différents noeuds pour estimer sa position.

- **Triangulation** [90] : Pour estimer la position en utilisant l'angle d'arrivée, deux angles sont requises. Le signal envoyé par le mobile au point M est reçu par deux points de référence (A_1, A_2) formant deux angles α et β . Revenant les positions de A_1 et A_2 , M peut calculer la distance d entre ces deux positions de référence (Figure 1.5).

Le point M estime sa position en utilisant le principe de la triangulation par la résolution de l'équation 1.2.

$$\begin{cases} x = x_1 + \frac{d \times \sin \alpha \times \sin \beta}{\sin(\alpha + \beta)} \\ y = \frac{d \times \cos \alpha \times \sin \beta}{\sin(\alpha + \beta)} \end{cases} \quad (1.2)$$

On se trouve face à deux problèmes majeurs en utilisant l'angle d'arrivée : (1) l'utilisation d'une série d'antennes pour détecter les angles, ce qui n'est pas pratique en termes de coût et énergie. (2) La mesure d'angle est affectée par plusieurs facteurs (les réflexions dues aux multitrajets et le bruit). Ce qui peut causer des larges erreurs dans l'estimation des angles.

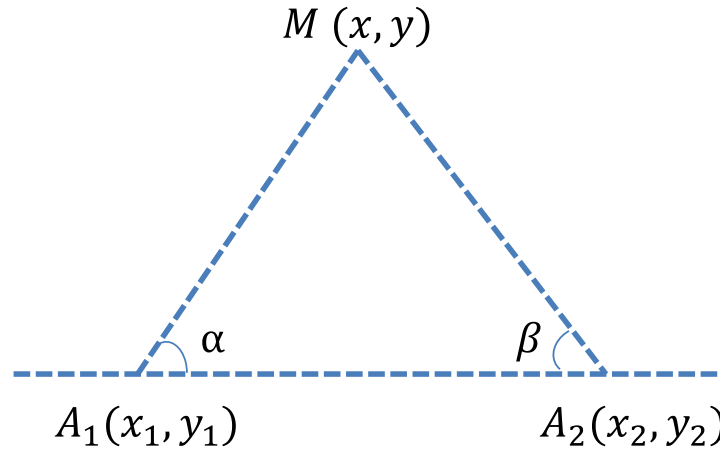


FIGURE 1.5: Localisation avec AoA -Triangulation-.

1.6.3 La méthode de fingerprinting

Pour remédier à certains points faibles des méthodes géométriques, une solution alternative à base de RSSI est mise en place ‘RSSI fingerprinting’ [91, 92]. Cette méthode comporte deux Phases. Une phase offline ou d’apprentissage (phase préliminaire) consiste à collecter des mesures de RSSI au niveau de différentes positions de référence. Pour une position donnée, l’ensemble des RSSI reçus des différents points d’accès (PAs) forment un fingerprint. L’ensemble des mesures collectées lors de cette phase associées aux coordonnées des positions correspondantes forment la base d’apprentissage nommée ‘radiomap’ (Figure 1.6). Des larges bases doivent être construites par plusieurs utilisateurs munis d’équipements mobiles différents afin de présenter diverses conditions de l’environnement intérieur, caractérisé par une forte dynamique. La phase online quant à elle débute par une évaluation de similarité. Cette similarité est calculée entre le fingerprint correspondant à la position de test à estimer et tous les fingerprints d’apprentissage. Ceci peut être fait par :

- Des méthodes déterministes : le système RADAR réalisé dans [93] utilise des méthodes déterministes pour comparer le fingerprint correspondant à l’objet d’intérêt et les fingerprints d’apprentissage.
- Une méthode probabiliste en accordant une probabilité de similarité à chaque fingerprint d’apprentissage.

A la fin de cette étape, on aura une valeur de similarité attribuée pour chaque fingerprint d’apprentissage. Pour estimer la position de test, une combinaison des positions d’apprentissage associées aux fingerprints correspondants aux valeurs de similarité les plus importantes est menée. On se trouve face à une phase offline lourde afin de créer une radiomap diversifiée et fiable et qui doit être mise à jour régulièrement selon la disponibilité du matériel pour tenir en compte les changements dans l’environnement où s’opère la localisation (ajout ou suppression de PAs, changement dans l’architecture des salles, etc). Cependant, cela rend

la localisation online plus robuste selon les algorithmes utilisés dont dépend la rapidité de la localisation.

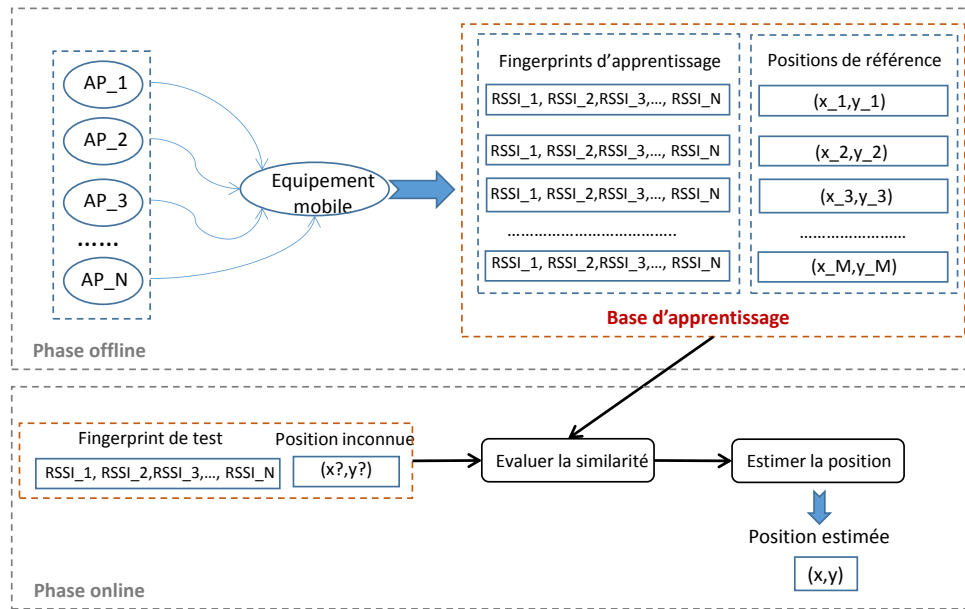


FIGURE 1.6: Les étapes de la technique de fingerprinting en considérant N points de référence et M positions d'apprentissage.

L'implémentation des solutions de localisation qui combinent la méthode de fingerprinting avec des méthodes de "machine learning (ML)" est très répandue. En effet, la préparation des données et le prétraitement sont assurés dans la phase offline (phase d'apprentissage), seule la tâche de prédiction est effectuée en temps réel. Par conséquent, pour calculer sa position, un noeud (capteur) interroge un modèle déjà entraîné, qui estime la position. De tels algorithmes transfèrent la complexité de calcul du processus de prédiction en temps réel vers la phase d'apprentissage. Ainsi, de telles solutions basées sur ML sont fortement recommandées pour les applications de localisation en temps réel. Cela est appuyé par l'efficacité des algorithmes de deep learning (DL) qui ont montré de très bonnes performances en localisation estimant soit les coordonnées de localisation, soit d'autres informations de localisation (identification de la pièce [94], identification de l'étage [95], identification de la zone [96, 97], etc). L'identification de zone ou de région contenant le capteur à localiser fait l'objet de quelques travaux récents [96, 97]. Cependant, ces solutions conçues ne sont pas flexibles dû au fait qu'elles nécessitent l'ajout des équipements spécifiques (des microphones ou des points de référence) aux sommets des régions ayant obligatoirement la forme carrée, ce qui est pratiquement impossible.

Nous rappelons que différentes approches basées sur les méthodes de DL, essentiellement Support Vector Machine (SVM) [98] et les réseaux de neurones, sont développées. Plusieurs types de réseaux de neurones ont été exploités, essentiellement les réseaux de neurones pro-

fonds (DNN) et leurs variantes : Multi Layer Perceptron (MLP), les réseaux de neurones récurrents (RNN), les réseaux de neurones convolutifs (CNN) [99, 100], etc. Pour faire face aux contraintes d'apprentissage réseau et réduire le nombre de paramètres (poids et biais) à apprendre et la complexité des réseaux de neurones (NN) traditionnels, CNN a été déployé. CNN est une classe de deep NN, largement utilisée. Elle réduit la complexité des NN traditionnels et le nombre de poids à apprendre grâce à sa structure de partage de poids. Cela signifie que CNN nécessite moins de paramètres à apprendre et peut apporter une meilleure performance et une meilleure robustesse. Une autre raison pour laquelle CNN suscite un intérêt important, est liée au fait qu'on a besoin de manipuler de grands ensembles de données requis par le NN traditionnel, afin d'éviter les problèmes de sur-apprentissage. Un autre défi dans la mise en oeuvre de systèmes de localisation basés sur les CNN est que ces réseaux ont une invariance par translation. Cette fonctionnalité coïncide avec la dépendance temporelle entre les fingerprints RSSI. Comme cette structure NN présente une invariance importante en translation, elle a été largement utilisée dans le traitement et la classification des images, obtenant ainsi un succès spectaculaire dans ce domaine. Ainsi, l'application de CNN sur des images RSSI présente un intérêt important pour la communauté de la localisation.

Dans [101], les chercheurs ont conçu un CNN pour une reconnaissance de l'activité des piétons, qui peut servir de repères pour la localisation en intérieur. Ici, les données unidimensionnelles issues des accéléromètres, des magnétomètres, des gyroscopes et des baromètres ont été considérées comme les entrées du réseau. Ce système nécessite l'usage de capteurs spécifiques et n'a pas pris en compte le problème de consommation d'énergie. Dans [101], les auteurs ont converti le signal sonore recueilli par un microphone en une carte spectrale pour l'intégrer dans un modèle CNN. Récemment, des systèmes de localisation conçus à partir de mesures RSSI ont été proposés. Dans [102], un classifieur hiérarchique a utilisé une combinaison de petits réseaux CNN qui fonctionnent ensemble pour prédire la position. Ce système traite des images RSSI en 2D, de taille $(N \times K)$, N étant le nombre de points d'apprentissage et K est le nombre de points d'accès (PAs). Les auteurs dans [103] identifient le bâtiment et l'étage, contenant l'utilisateur à localiser, en utilisant les RSSI obtenus à partir des PAs voisins. A partir d'un RSSI fingerprint associé à un point d'apprentissage, une image 2D est créée, ajoutant quelques valeurs muettes. (par exemple : une image 2D de dimension (23×23) est construite à partir d'un RSSI fingerprint (520×1) où 520 correspond au nombre de PAs, ajoutant 9 valeurs muettes). Un CNN hiérarchique a été proposé dans [18] utilisant des images RSSI combinant des données WiFi et magnétiques dans une seule image. La branche WiFi et la branche magnétique produisent deux vecteurs de prédictions différentes. Ensuite, ces vecteurs vont être combinés en tant qu'entrée d'une branche unie pour estimer la position de l'utilisateur. Un framework a été implémenté dans [100] en utilisant CNN pour identifier le bâtiment et l'étage, puis un DNN a été créé pour estimer la position. Dans ce travail, les images 2D considérées, dont chacune correspond à un point d'apprentissage donné, sont formées par les mesures RSSI reçues par ce point de différents PAs à différents instants.

Les images radio 3D sont très peu utilisées dans le contexte de localisation. Leur usage est lié essentiellement à la localisation des robots car plusieurs types de capteurs sont intégrés dans un robot (caméra, laser, compteur kilométrique, etc.) [103–105]. Par conséquent, chaque plan de l'image radio contenait les données reçues d'un capteur spécifique. Outre la localisation des robots, les images radio 3D peuvent être utilisées dans des systèmes explo-

rant différents types de données. Par exemple, les auteurs dans [18] ont exploré les données RSSI accompagnées d'informations magnétiques et d'accélération. Basé sur CSI, les images radio 3D peuvent être générées tel que développé dans [106], où une matrice CSI d'une antenne est considérée comme le plan rouge, vert et bleu de l'image. Par conséquent, l'image a été construite en combinant trois canaux de CSI.

1.7 Les critères d'évaluation de performances

Dans cette thèse, on utilisera la méthode de trilatération ainsi que la méthode de fingerprinting en les adaptant à des applications IoT et leurs contraintes en architecture (centralisée ou pas) et en essayant d'améliorer les performances de chacune en utilisant des algorithmes classiques et des algorithmes avancés. Pour évaluer les performances d'un système de localisation, plusieurs critères peuvent être considérés.

- Précision de localisation : Elle renseigne sur la fiabilité du système. Quelle est la probabilité d'obtenir une bonne estimation de position.
- Intervalle de mise à jour : l'intervalle de mise à jour indique la fréquence à laquelle les informations de localisation sont mises à jour et reflète la consommation d'énergie possible. Une solution simple pour réduire la consommation d'énergie du système de localisation est d'augmenter l'intervalle de mise à jour.
- Coût de calcul et complexité : la complexité est un critère important pour évaluer une méthode de localisation. Plus de coût de calcul signifie plus de consommation d'énergie et plus de temps de réponse.
- Infrastructure : Signifie s'il faut ajouter une infrastructure supplémentaire pour le déploiement du système ou non, ce qui est important au niveau budgétaire. La réutilisation et la maintenance de l'infrastructure sont également significatifs pour le déploiement.
- Calcul hors ligne : Certaines méthodes de localisation en intérieur nécessitent un calcul ou une phase offline nécessitant un travail intensif et un temps de déploiement plus long. Cette exigence augmente le coût et le temps de déploiement ainsi que les coûts de maintenance.
- Durée de localisation : temps nécessaire au système pour une localisation en intérieur sans fil. Pour les méthodes prenant en charge la localisation pour un objet immobile, le temps requis pour la localisation peut être très rapide.

Dans nos travaux, nous avons considéré des environnements intérieurs avec du réseau WiFi où nous exploitons l'infrastructure existante sans coût supplémentaire. L'intervalle de mise à jour n'a pas fait partie de nos études puisqu'il change d'une application à une autre et d'un service à un autre selon les besoins et les attentes requises. Nous avons considéré la précision de localisation, la complexité et le temps requis pour la localisation comme critères d'évaluation de performances et de comparaison de différents systèmes proposés et implémentés.

1.8 Objectifs de la thèse

Le contexte et les attentes d'un système de localisation changent d'une application à une autre (niveau de précision requis, complexité, temps de réponse, etc.). De ce fait et pour satisfaire aux différentes hypothèses et scénarios pouvant être confrontés en intérieur, nous avons proposé des solutions de localisation adaptées à chaque architecture IoT.

- Solutions de localisation pour une architecture distribuée (un objet communique avec tous les objets : PAs et objets simples) où on exploite la trilatération. Au niveau de cette technique, ce qui cause problème est le nombre de PAs détectés. Pour augmenter ce nombre et fournir une information de distance plus riche et plus fiable, nous avons utilisé la technique de complétion de matrice.
- Solutions de localisation pour une architecture centralisée (un objet communique seulement avec les PAs) où on utilise la méthode de fingerprinting. La comparaison de similarité entre le fingerprint associé à l'objet à localiser et les fingerprints d'apprentissage est une étape assez importante et délicate dans le processus de localisation. Pour cela, nous avons proposé une étude comparative pour évaluer cette similarité sans modèle. Et nous sommes intervenus au niveau du fingerprinting en utilisant, soit un modèle linéaire, qui permet de trouver une relation mathématique entre les puissances reçues et le vecteur de position, soit un modèle prédictif de type CNN.

1.9 Conclusion

Dans ce chapitre, nous avons présenté le contexte dans lequel se présente la thématique traitée par cette thèse. Nous avons défini les notions principales utilisées dans le reste de cette thèse et nous avons justifié les choix faits. Dans le reste du rapport, nous allons exploiter la méthode de fingerprinting et la trilatération afin d'améliorer les performances de chacune. Pour cela, des environnements dotés de la technologie sans fil seront étudiés et la puissance du signal reçu sera exploitée.

Chapitre 2

Solution de localisation pour une architecture distribuée

Comme mentionné dans le premier chapitre, il est difficile d'acquérir toutes les distances par paire entre les capteurs et obtenir une matrice de distances, appelée Matrice de distances euclidiennes (MDE) [25], entièrement connue. Ceci est dû à la limitation de la portée d'une communication radio et aux effets des trajets multiples. Pour fournir une information de distance fiable à chaque noeud capteur, la technique de complétion de matrice est ainsi proposée. Son rôle est de récupérer une MDE complète à partir d'une matrice incomplète à entrées partiellement connues. Plusieurs algorithmes et approches ont été proposés et étudiés pour approcher les distances manquantes dans les systèmes basés sur les mesures RSSI. Dans le reste de ce chapitre, on détaille notre approche et on évalue ses performances.

2.1 Système modèle

Nous donnons une description détaillée du modèle proposé utilisé pour localiser un noeud mobile ou fixe dans un environnement intérieur. L'emplacement approximatif de chaque noeud est obtenu en utilisant un processus d'optimisation et la technique de trilatération basée sur les distances.

On considère le réseau illustré dans la Figure 2.1 dans lequel n noeuds communiquent entre eux. m noeuds sont des points de références avec des positions connues et sont appelés ancres A_j et $j = 1 \dots m$. Les $n - m$ noeuds restants ont des positions inconnues (à localiser), ils sont appelés utilisateurs et notés U_i avec $i = 1 \dots n - m$.

Chaque noeud mesure les RSSI reçus de tous les autres noeuds du réseau qu'il reçoit. Ainsi un vecteur des RSSI est construit dans lequel certaines valeurs ne sont pas renseignées et correspondent aux RSSI qui sont de valeur inférieure à la sensibilité du noeud (c'est un point d'interrogation en jaune sur Figure 2.1).

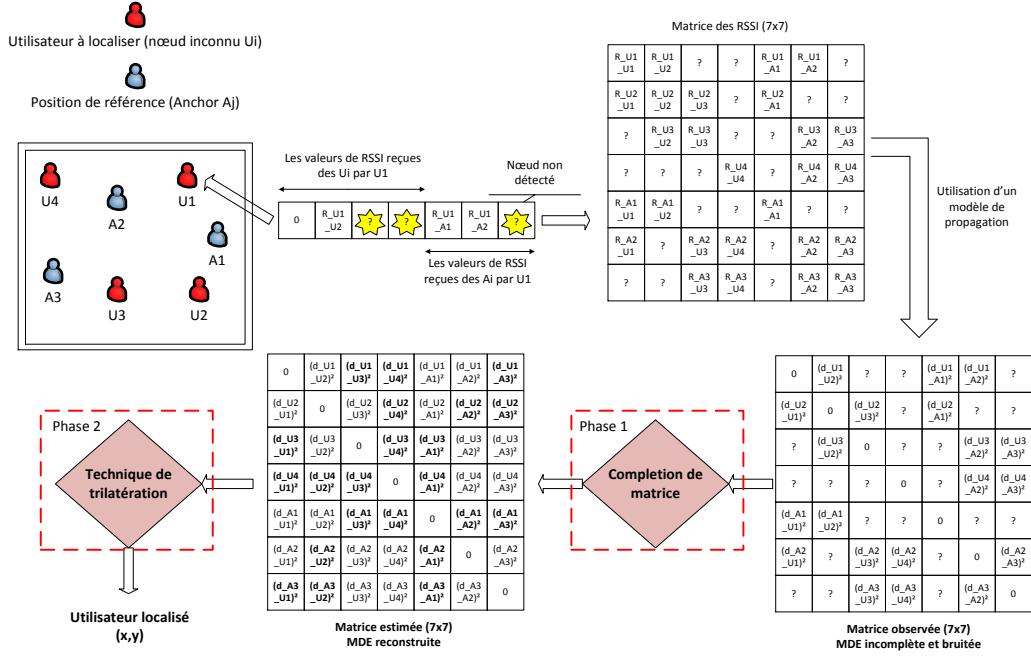


FIGURE 2.1: Les détails du framework développé à base de la complétion de matrice en considérant un réseau contenant 7 noeuds capteurs dont 3 sont des anchors.

A partir des valeurs RSSI, la matrice de distances euclidiennes contenant les distances entre chaque paire de noeuds \mathbf{X} est construite en utilisant le modèle de propagation donné par équation 2.2. Les valeurs de RSSI sont calculées en dB comme suit :

$$pr_{ij} = pe - pl_{ij} + a_{\sigma} \quad (2.1)$$

avec pe qui est la puissance d'émission, a_{σ} est une variable Gaussienne aléatoire qui décrit les effets aléatoires du shadowing, pl_{ij} est l'affaiblissement du trajet. Pour exprimer cet affaiblissement en dB [107], on utilise le modèle de propagation radio suivant :

$$pl_{ij} = pl_0 + 20\log_{10}(f) + 10\alpha\log_{10}\left(\frac{d}{d_0}\right) \quad (2.2)$$

Dû à la limitation de la portée de communication radio, la matrice de RSSI est partiellement connue. Ainsi, la matrice \mathbf{X} est incomplète (seul un petit nombre des entrées de \mathbf{X} est disponible) et peut être affectée par le bruit. Cette matrice incomplète ne peut pas servir efficacement pour la localisation. D'où, la nécessité de la compléter afin de trouver les positions des utilisateurs. Définissons la matrice \mathbf{X}_{true} qui est la MDE complète contenant les distances réelles. Par conséquent, le problème est : comment récupérer les éléments inconnus de \mathbf{X} en partant d'un petit nombre d'entrées connues de \mathbf{X} .

Supposons que m est le nombre de noeuds de référence de positions connues, appelées 'anchors'. En revanche, $(n - m)$ est le nombre de capteurs dont les positions sont inconnues.

n est le nombre total de noeuds capteurs (anchors et noeuds inconnus) placés dans l'environnement intérieur. U_i est le $i^{\text{ième}}$ noeud inconnu avec $i = 1, 2, \dots, (n - m)$. A_j est le $j^{\text{ième}}$ anchor où $j = 1, 2, \dots, m$. \mathbf{X} de dimension $(n \times n)$ est la matrice de distances euclidiennes. Elle peut être présentée comme suit :

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{bmatrix} \quad (2.3)$$

Où \mathbf{X}_{11} est la sous-matrice de distances entre chaque paire de noeuds inconnus, de dimension $(n - m) \times (n - m)$. \mathbf{X}_{12} et \mathbf{X}_{21} , où $\mathbf{X}_{12} = \mathbf{X}_{21}^T$, sont les sous-matrices de distances entre les anchors et les noeuds inconnus. \mathbf{X}_{22} est la sous-matrice de distances entre chaque paire d'anchors. \mathbf{X}_{11} , \mathbf{X}_{12} et \mathbf{X}_{21} sont obtenues à partir des mesures RSSI utilisant un modèle de propagation. \mathbf{X}_{22} est obtenue en calculant la distance exacte entre chaque paire des anchors. $d_{A_i A_j} = \|C_{A_i} - C_{A_j}\|_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ est la distance entre l'anchor i (A_i) et l'anchor j (A_j). $C_{A_i} \in R^2$ sont les coordonnées de l'anchor i où $C_{A_i} = (x_i, y_i)$. Les distances par paires sont dépendantes, cela est dû à la dépendance des RSSI dans les environnements intérieurs. La matrice de distances doit donc avoir un rang faible. C'est cette propriété qui motive l'utilisation de la complétion de matrice.

2.2 Localisation à base de la complétion de matrice

Dans cette section, nous détaillons les différentes étapes effectuées par un utilisateur (noeud capteur) pour se localiser. Après avoir complété la matrice de distances, la technique de trilatération est appliquée afin d'estimer la position. Avant de commencer ce processus, nous proposons une formulation du problème de complétion de matrice comme étant un problème de minimisation du rang de la matrice de distances. En effet, comme la communication entre les noeuds du réseau se fait dans les mêmes conditions de propagation, les mesures de RSSI sont corrélées. La relation entre RSSI et les distances étant linéaire, les lignes et les colonnes de la matrice sont dépendantes d'où le rang faible de cette dernière.

2.2.1 Formulation du problème de la complétion de matrice

Notre objectif est de reconstruire la matrice de distances complète à partir de données incomplètes et bruitées. Le problème de récupération d'une matrice de rang faible à partir d'un petit nombre d'entrées connues est défini comme suit :

$$\begin{cases} \min_{\hat{\mathbf{X}}} & rank(\hat{\mathbf{X}}) \\ s.t. & \hat{x}_{ij} = x_{ij} \quad \text{où} \quad (i, j) \in \omega \end{cases} \quad (2.4)$$

ω est l'ensemble des entrées connues.

En raison de la non convexité de la fonction rang de la matrice, le problème de l'équation 2.4 ne peut pas être résolu analytiquement. Inspiré de la théorie de compressed sensing (CS), Candes et Recht ont proposé de remplacer la fonction de rang dans l'équation 2.4 par la norme nucléaire [108]. Le modèle convexe est reformulé en

$$\begin{cases} \min_{\hat{\mathbf{X}}} & \|\hat{\mathbf{X}}\|_* \\ \text{s.t. } & \hat{x}_{ij} = x_{ij} \quad \text{où } (i, j) \in \omega \end{cases} \quad (2.5)$$

$\|\hat{\mathbf{X}}\|_*$ est la somme des valeurs singulières de $\hat{\mathbf{X}}$ (c'est à dire $\|\hat{\mathbf{X}}\|_* = \sum_{j=1}^n s_j$). Considérons l'hypothèse de rang faible ($r \ll n$) et en tenant compte du fait que les observations sont généralement affectées par un bruit, le modèle de complétion de la matrice peut être défini comme suit

$$\min_{\hat{\mathbf{X}}} \lambda \times \|\hat{\mathbf{X}}\|_* + \|\mathbf{H} \odot (\hat{\mathbf{X}} - \mathbf{X})\|_F^2 \quad (2.6)$$

λ est un paramètre ajustable et $\|\cdot\|_F$ est la norme de Frobenius. $\|\hat{\mathbf{X}}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |\hat{x}_{ij}|^2}$. \mathbf{H} est la matrice dont les entrées sont

$$h_{ij} = h_{ji} = \begin{cases} 1 & \text{si } (i, j) \in \omega \\ 0 & \text{sinon} \end{cases} \quad (2.7)$$

La fonction coût peut être définie comme suit

$$J(\hat{\mathbf{X}}) = f(\hat{\mathbf{X}}) + \lambda \times l(\hat{\mathbf{X}}) \quad (2.8)$$

Où $f(\hat{\mathbf{X}}) = \|\mathbf{H} \odot (\hat{\mathbf{X}} - \mathbf{X})\|_F^2$ et $l(\hat{\mathbf{X}}) = \|(1 - \mathbf{H}) \odot \hat{\mathbf{X}}\|_*$. Le problème d'optimisation défini peut être résolu efficacement en utilisant la méthode itérative de la descente de gradient et ses variantes. L'algorithme développé est décrit dans tableau 2.1. Où $\mathbf{V}^{(t)}$ est la matrice de gradient de la fonction coût $J(\hat{\mathbf{X}})$ et utilisée pour mettre à jour la matrice de distances, indice t fait référence au nombre d'itérations de mise à jour. De nombreuses approches, détaillées dans la sous section suivante, ont été adoptées pour assurer la mise à jour matricielle $\mathbf{V}^{(t)}$.

TABLE 2.1: L'algorithme décrit en 4 étapes.

1. Entrées : \mathbf{X} , \mathbf{H} , nombre max d'itérations

2. Initialisation : $\hat{\mathbf{X}}^{(0)}$, t

3. Tant que $t < \text{nombre max d'itérations}$

Mettre à jour $\hat{\mathbf{X}}$:

$$\hat{\mathbf{X}}^{(t+1)} = \hat{\mathbf{X}}^{(t)} - \mathbf{V}^{(t)} \quad (2.9)$$

$t = t + 1$

Fin

4. Retourner $\hat{\mathbf{X}}$

2.2.2 Complétion de matrice : Optimisation avec Gradient Descendant (GD) et ses variantes

Dans ce qui suit, nous détaillons les différentes méthodes d'optimisation largement utilisées par la communauté de deep learning pour mettre à jour la matrice de localisation $\hat{\mathbf{X}}$: Gradient Descendant (GD), Nesterov Acceleration Gradient (NAG), Adaptatif Gradient (Adagrad), Root Mean Square Propagation (RMSProp), Adadelta et Adaptative Moment Estimation (Adam). Nous ne considérons pas la classe des algorithmes très coûteux en terme de calcul pour les données volumineuses, par exemple la méthode de Newton du second ordre.

1. Gradient Descendant (GD)

La descente de gradient est une méthode itérative qui vise à trouver un minimum local de fonctions de coût différentiables [109]. C'est l'algorithme d'optimisation de premier ordre le plus répandu en machine learning et deep learning. GD est basé sur la mise à jour de chaque élément de la matrice $\hat{\mathbf{X}}^{(t)}$ pour l'optimisation de la fonction objectif $J(\hat{\mathbf{X}}^{(t)})$. Le nouveau paramètre $\mathbf{V}^{(t)}$ peut être ajusté comme suit :

$$\mathbf{V}^{(t)} = \alpha \nabla(J(\hat{\mathbf{X}}^{(t)})) \quad (2.10)$$

α est le taux d'apprentissage dont la valeur varie entre 0 et 1. $\nabla(J(\hat{\mathbf{X}}^{(t)}))$ est le gradient de la fonction coût par rapport à la matrice de paramètres. Il peut être calculé comme suit :

$$\nabla(J(\hat{\mathbf{X}}^{(t)})) = \nabla(f(\hat{\mathbf{X}}^{(t)})) + \lambda \times \nabla(l(\hat{\mathbf{X}}^{(t)})) \quad (2.11)$$

Où

$$\begin{aligned} \nabla(f(\hat{\mathbf{X}}^{(t)})) &= \mathbf{U}^{(t)} = \nabla(\|\mathbf{H} \odot (\hat{\mathbf{X}}^{(t)} - \mathbf{X}^{(t)})\|_F^2) \\ &= 2 \times \mathbf{H} \odot (\hat{\mathbf{X}}^{(t)} - \mathbf{X}^{(t)}) \end{aligned} \quad (2.12)$$

$\nabla(l(\hat{\mathbf{X}}^{(t)}))$ est calculé comme suit :

$$\begin{aligned} \nabla(l(\hat{\mathbf{X}}^{(t)})) &= \mathbf{W}^{(t)} = \nabla(\|(1 - \mathbf{H}) \odot \hat{\mathbf{X}}^{(t)}\|_*) \\ &= (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t)} \cdot ((\hat{\mathbf{X}}^{(t)})^T \cdot \hat{\mathbf{X}}^{(t)})^{-0.5}) \end{aligned} \quad (2.13)$$

Une sorte de régularisation est requise pour éviter l'inversion de zéro, par exemple,

$$\nabla(l(\hat{\mathbf{X}}^{(t)})) = \nabla(\|(1 - \mathbf{H}) \odot \hat{\mathbf{X}}^{(t)}\|_*) = (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t)} \cdot ((\hat{\mathbf{X}}^{(t)})^T \cdot \hat{\mathbf{X}}^{(t)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \quad (2.14)$$

Où ϵ est un paramètre de régularisation et \mathbf{I}_n est la matrice identité de dimension $(n \times n)$.

$$\nabla(J(\hat{\mathbf{X}}^{(t)})) = 2 \times \mathbf{H} \odot (\hat{\mathbf{X}}^{(t)} - \mathbf{X}^{(t)}) + \lambda \times (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t)} \cdot ((\hat{\mathbf{X}}^{(t)})^T \cdot \hat{\mathbf{X}}^{(t)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \quad (2.15)$$

On obtient

$$\mathbf{V}^{(t)} = \alpha \times (2 \times \mathbf{H} \odot (\hat{\mathbf{X}}^{(t)} - \mathbf{X}^{(t)}) + \lambda \times (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t)} \cdot ((\hat{\mathbf{X}}^{(t)})^T \cdot \hat{\mathbf{X}}^{(t)} + \epsilon \times \mathbf{I}_n)^{-0.5})) \quad (2.16)$$

En utilisant la descente de gradient classique, les entrées connues sont très bien estimées, c'est pourquoi elle sera toujours utilisée pour estimer $\mathbf{U}^{(t)}$. Pour estimer $\mathbf{W}^{(t)}$, nous proposons d'utiliser les algorithmes cités ci-après.

2. Nesterov Accelerated Gradient (NAG)

Une deuxième variante couramment utilisée est l'accélération de Nesterov, publiée dans [110] et s'inscrit dans la même veine que l'idée du Momentum [111]. Elle a la même intuition en utilisant l'historique du gradient, mais elle calcule le gradient par rapport aux valeurs futures approximatives des paramètres au lieu des paramètres actuels. Pour mettre à jour $\mathbf{W}^{(t)}$ avec NAG, nous utilisons l'équation suivante.

$$\mathbf{W}^{(t)} = \mu \times \mathbf{W}^{(t-1)} + \lambda \times \nabla(l(\hat{\mathbf{X}}^{(t)} - \mu \times \mathbf{W}^{(t-1)})) \quad (2.17)$$

Où

$$\mathbf{W}^{(t-1)} = \nabla(l(\hat{\mathbf{X}}^{(t-1)})) = (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t-1)} \cdot ((\hat{\mathbf{X}}^{(t-1)})^T \cdot \hat{\mathbf{X}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \quad (2.18)$$

Et

$$\nabla(l(\hat{\mathbf{X}}^{(t)} - \mu \times \mathbf{W}^{(t-1)})) = \nabla(l(\hat{\mathbf{X}}^{(t)} - \mu \times (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t-1)} \cdot ((\hat{\mathbf{X}}^{(t-1)})^T \cdot \hat{\mathbf{X}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}))) \quad (2.19)$$

On suppose que

$$\hat{\mathbf{Y}}^{(t)} = \hat{\mathbf{X}}^{(t)} - \mu \times (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t-1)} \cdot ((\hat{\mathbf{X}}^{(t-1)})^T \cdot \hat{\mathbf{X}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \quad (2.20)$$

Puis,

$$\nabla(l(\hat{\mathbf{Y}}^{(t)})) = (1 - \mathbf{H}) \odot (\hat{\mathbf{Y}}^{(t)} \cdot ((\hat{\mathbf{Y}}^{(t-1)})^T \cdot \hat{\mathbf{Y}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \quad (2.21)$$

On obtient

$$\begin{aligned} \mathbf{W}^{(t)} = & \mu \times (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t-1)} \cdot ((\hat{\mathbf{X}}^{(t-1)})^T \cdot \hat{\mathbf{X}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \\ & + \lambda \times (1 - \mathbf{H}) \odot (\hat{\mathbf{Y}}^{(t)} \cdot ((\hat{\mathbf{Y}}^{(t-1)})^T \cdot \hat{\mathbf{Y}}^{(t-1)} + \epsilon \times \mathbf{I}_n)^{-0.5}) \end{aligned} \quad (2.22)$$

Le principal avantage de la méthode NAG par rapport à la descente de gradient classique est lié au fait qu'une mise à jour anticipée nous empêche d'aller trop vite et conduit à une réactivité accrue à la fonction coût.

3. Adaptive Gradient (Adagrad)

Duchi et al. [112] ont introduit l'algorithme d'Adagrad dans le contexte de la méthode du gradient projeté. Adagrad a pour objectif d'adapter le taux d'apprentissage aux paramètres mis à jour, introduisant un faible taux d'apprentissage (c'est-à-dire des mises à jour plus petites) lorsque la mémoire des gradients carrés est élevée et des mises à jour plus grandes. La règle de mise à jour de Adagrad est la suivante :

- $\mathbf{E}^{(t)}$ est le gradient de la fonction coût par rapport au paramètre $\hat{\mathbf{X}}^{(t)}$

$$\begin{aligned}\mathbf{E}^{(t)} &= \frac{\delta l(\hat{\mathbf{X}}^{(t)})}{\delta \hat{\mathbf{X}}^{(t)}} \\ &= (1 - \mathbf{H}) \odot (\hat{\mathbf{X}}^{(t)} \cdot ((\hat{\mathbf{X}}^{(t)})^T \cdot \hat{\mathbf{X}}^{(t)} + \epsilon \times \mathbf{I}_n)^{-0.5})\end{aligned}\quad (2.23)$$

- La mémoire des gradients carrés au fil du temps est calculée

$$\mathbf{G}^{(t)} = \sum_{i=1}^t (\mathbf{E}^{(i)})^2 \quad (2.24)$$

- Le taux d'apprentissage α est modifié à chaque t pour chaque paramètre $\hat{\mathbf{X}}^{(t)}$ en se basant sur la somme des carrés des gradients calculés pour $\hat{\mathbf{X}}^{(t)}$ au fil du temps t .

$$\mathbf{W}^{(t)} = \frac{1}{\sqrt{\mathbf{G}^{(t)} + \epsilon \times \mathbf{I}_n}} \odot \mathbf{E}^{(t)} \quad (2.25)$$

Où ϵ est un terme de régularisation utilisé pour éviter la division par zéro. Il est à noter qu'il n'est pas nécessaire d'ajuster le taux d'apprentissage.

4. Root Mean Square Propagation (RMSProp)

Tieleman et al. [113] ont introduit cet algorithme en 2012. Il est décrit dans les 2 étapes suivantes. Au lieu d'utiliser la mémoire de tous les gradients carrés, RMSProp utilise uniquement l'historique des gradients récents calculés dans une fenêtre de temps limitée.

- Nous calculons la moyenne locale des précédents $(\mathbf{E}^{(t)})^2$

$$\tilde{\mathbf{G}}^{(t)} = \rho \times \mathbf{G}^{(t-1)} + (1 - \rho) \times (\mathbf{E}^{(t)})^2 \quad (2.26)$$

- Après, on applique la mise à jour

$$\mathbf{W}^{(t)} = \frac{1}{\sqrt{\tilde{\mathbf{G}}^{(t)} + \epsilon \times \mathbf{I}_n}} \odot \mathbf{E}^{(t)} \quad (2.27)$$

5. Adadelta

Adadelta a été introduit en 2012 par Zeiler [113]. Il vise à contourner la faiblesse d'Adagrad qui consiste en son taux d'apprentissage en décroissance agressive causée par l'accumulation de tous les gradients carrés antérieurs au dénominateur. Adadelta utilise uniquement les gradients passés récents calculés dans un temps limité (c'est-à-dire pas tout l'historique). En outre, Adadelta utilise un terme accélérateur en prenant en compte les mises à jour passées (comme dans Momentum). La règle de mise à jour de Adadelta est la suivante :

- On calcule le gradient $\mathbf{E}^{(t)}$ comme dans l'équation (2.23).
- On calcule la moyenne locale $\tilde{\mathbf{G}}^{(t)}$ des $(\mathbf{E}^{(t)})^2$ antérieurs
- On calcule le nouveau terme accumulant les mises à jour antérieures (Momentum : terme d'accélération)

$$\mathbf{Z}^{(t)} = \rho \times \mathbf{Z}^{(t-1)} + (1 - \rho) \times (\mathbf{W}^{(t-1)})^2 \quad (2.28)$$

- Puis, la mise à jour est assurée

$$\mathbf{W}^{(t)} = \frac{\sqrt{\mathbf{Z}^{(t)} + \epsilon \times \mathbf{I}_n}}{\alpha \sqrt{\tilde{\mathbf{G}}^{(t)} + \epsilon \times \mathbf{I}_n}} \odot \mathbf{E}^{(t)} \quad (2.29)$$

6. Adaptive Moment Estimation (Adam)

Une autre méthode d'optimisation permettant de calculer un taux d'apprentissage adaptatif pour chaque paramètre est introduite par Kingma et Ba dans [114]. Adam utilise le premier et le deuxième moments des gradients et a de fortes similitudes avec Adadelta. En effet, cette méthode utilise le deuxième moment de gradient dans le dénominateur et un terme de momentum. La mise à jour conduite par Adam comprend les étapes suivantes.

- Calculer le deuxième moment de gradient avec une accumulation locale (Adadelta/RMSProp)

$$\mathbf{N}^{(t)} = \beta_1 \times \mathbf{N}^{(t-1)} + (1 - \beta_1) \times (\mathbf{E}^{(t)})^2 \quad (2.30)$$

- Calculer le premier moment de gradient

$$\mathbf{M}^{(t)} = \beta_2 \times \mathbf{M}^{(t-1)} + (1 - \beta_2) \times \mathbf{E}^{(t)} \quad (2.31)$$

- Calculer les estimations du premier et du deuxième moment corrigés

$$\hat{\mathbf{N}}^{(t)} = \frac{\mathbf{N}^{(t)}}{1 - \beta_1} \quad (2.32)$$

$$\hat{\mathbf{M}}^{(t)} = \frac{\mathbf{M}^{(t)}}{1 - \beta_2} \quad (2.33)$$

- Mettre à jour les paramètres

$$\mathbf{W}^{(t)} = \frac{\hat{\mathbf{M}}^{(t)}}{\sqrt{\hat{\mathbf{N}}^{(t)} + \epsilon \times \mathbf{I}_n}} \quad (2.34)$$

2.2.3 Localisation à base de trilatération

Après avoir complété la matrice de distances Euclidiennes, un processus de trilatération est adopté par chaque noeud inconnu afin d'estimer son emplacement [71, 72]. Pour cela, une combinaison de distances estimées est requise. Lors de process de combinaison, nous utilisons uniquement la sous-matrice de distances estimées \hat{X}_{12} qui contient les distances entre les noeuds inconnus et tous les anchors. La complétion de toute la matrice apporte plus d'informations sur les distances. Ces dernières sont estimées les unes des autres étant donné que les colonnes sont dépendantes (matrice de rang faible).

$$\hat{\mathbf{X}}_{12} = \begin{bmatrix} d_{11}^2 & d_{12}^2 & \cdot & \cdot & d_{1m}^2 \\ d_{21}^2 & d_{22}^2 & \cdot & \cdot & d_{2m}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{(n-m)1}^2 & d_{(n-m)2}^2 & \cdot & \cdot & d_{(n-m)m}^2 \end{bmatrix} \quad (2.35)$$

Soient (\hat{x}_j, \hat{y}_j) les coordonnées estimées d'un noeud inconnu ; $\hat{C}_j = [\hat{x}_j, \hat{y}_j]^T$. (x_i, y_i) , $(i = 1, 2, \dots, m)$ sont les coordonnées des ancres. $d_{ji}^2 = (x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2$ où $j = (1, 2, \dots, n-m)$. Pour un noeud inconnu U_j , nous avons

$$\begin{cases} d_{j1}^2 = (x_1 - \hat{x}_j)^2 + (y_1 - \hat{y}_j)^2 \\ d_{j2}^2 = (x_2 - \hat{x}_j)^2 + (y_2 - \hat{y}_j)^2 \\ d_{j3}^2 = (x_3 - \hat{x}_j)^2 + (y_3 - \hat{y}_j)^2 \\ \vdots \\ d_{jm}^2 = (x_m - \hat{x}_j)^2 + (y_m - \hat{y}_j)^2 \end{cases} \quad (2.36)$$

En développant l'équation 2.37 par rapport aux coordonnées des ancres, nous obtenons

$$\begin{cases} d_{j1}^2 = x_1^2 + \hat{x}_j^2 - 2x_1\hat{x}_j + y_1^2 + \hat{y}_j^2 - 2y_1\hat{y}_j \\ d_{j2}^2 = x_2^2 + \hat{x}_j^2 - 2x_2\hat{x}_j + y_2^2 + \hat{y}_j^2 - 2y_2\hat{y}_j \\ d_{j3}^2 = x_3^2 + \hat{x}_j^2 - 2x_3\hat{x}_j + y_3^2 + \hat{y}_j^2 - 2y_3\hat{y}_j \\ \vdots \\ d_{jm}^2 = x_m^2 + \hat{x}_j^2 - 2x_m\hat{x}_j + y_m^2 + \hat{y}_j^2 - 2y_m\hat{y}_j \end{cases} \quad (2.37)$$

et

$$\begin{cases} d_{j2}^2 - d_{j1}^2 = x_2^2 - x_1^2 + y_2^2 - y_1^2 - 2(\hat{x}_j x_2 + \hat{y}_j y_2 - \hat{x}_j x_1 - \hat{y}_j y_1) \\ d_{j3}^2 - d_{j1}^2 = x_3^2 - x_1^2 + y_3^2 - y_1^2 - 2(\hat{x}_j x_3 + \hat{y}_j y_3 - \hat{x}_j x_1 - \hat{y}_j y_1) \\ \vdots \\ d_{jm}^2 - d_{j1}^2 = x_m^2 - x_1^2 + y_m^2 - y_1^2 - 2(\hat{x}_j x_m + \hat{y}_j y_m - \hat{x}_j x_1 - \hat{y}_j y_1) \end{cases} \quad (2.38)$$

Considérons que $k_i^2 = x_i^2 + y_i^2$, l'équation 2.38 devient

$$\begin{cases} d_{j2}^2 - d_{j1}^2 = k_2^2 - k_1^2 - 2(\hat{x}_j x_2 + \hat{y}_j y_2 - \hat{x}_j x_1 - \hat{y}_j y_1) \\ d_{j3}^2 - d_{j1}^2 = k_3^2 - k_1^2 - 2(\hat{x}_j x_3 + \hat{y}_j y_3 - \hat{x}_j x_1 - \hat{y}_j y_1) \\ \vdots \\ d_{jm}^2 - d_{j1}^2 = k_m^2 - k_1^2 - 2(\hat{x}_j x_m + \hat{y}_j y_m - \hat{x}_j x_1 - \hat{y}_j y_1) \end{cases} \quad (2.39)$$

Ce système d'équations peut être exprimé sous forme matricielle

$$\mathbf{B}\hat{C}_j = \mathbf{P} \quad (2.40)$$

avec

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} k_2^2 - k_1^2 - d_{j2}^2 + d_{j1}^2 \\ k_3^2 - k_1^2 - d_{j3}^2 + d_{j1}^2 \\ \dots\dots \\ \dots\dots \\ k_m^2 - k_1^2 - d_{jm}^2 + d_{j1}^2 \end{bmatrix} ; \mathbf{B} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ x_m - x_1 & y_m - y_1 \end{bmatrix}$$

Donc, les coordonnées estimées de l'utilisateur U_j sont calculées comme suit

$$\hat{C}_j = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \quad (2.41)$$

2.3 Evaluation des performances

Dans cette section, nous ajustons par simulations les différents paramètres utilisés. L'utilisation bénéfique de la complétion de matrice pour améliorer la précision de la localisation assurée par trilatération est justifiée expérimentalement. Nous étudions également les performances d'estimation des distances inconnues, de la précision de localisation et de la complexité de calcul associée à chaque algorithme étudié.

2.3.1 Configuration de l'environnement de simulation

Nous considérons un réseau sans fil de 45 noeuds dont 10 ancres et 35 utilisateurs inconnus, placés dans une zone de 400 m² (c'est-à-dire 20 m × 20 m). Son architecture est illustrée dans la figure 2.2.

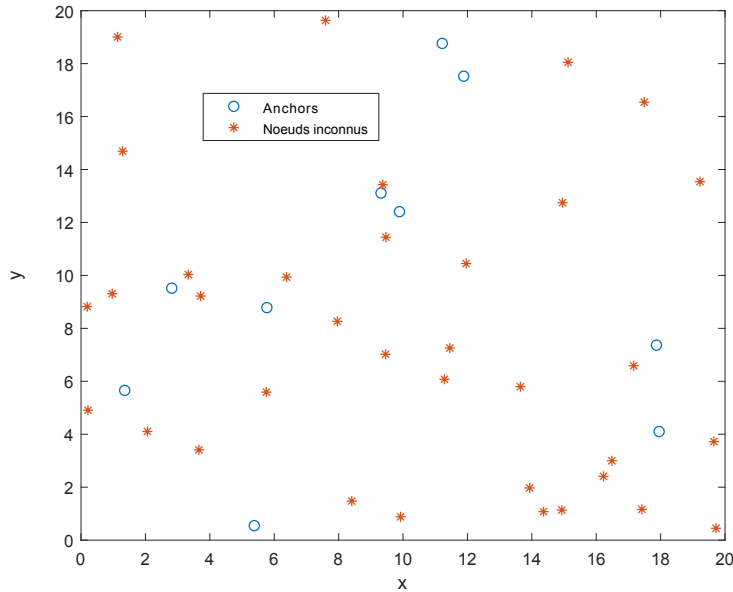


FIGURE 2.2: Configuration du réseau de capteurs sans fil.

Les noeuds capteurs (anchors et noeuds inconnus) sont placés de manière aléatoire dans la zone étudiée. La précision de tous les algorithmes utilisés est étudiée à travers plusieurs réalisations d'environnement. Afin de simplifier la présentation du document et sans perte de généralité, nous présentons les résultats uniquement pour un environnement de test, comme illustré à la figure 2.2. Les résultats de la simulation sont cohérents avec d'autres environnements. 10 mesures de RSSI (pr_{ij}) reçues du capteur i ($i = 1, 2, 3, \dots, n$) sont prises à chaque position de capteur j ($j = 1, 2, 3, \dots, n$) pour chaque valeur de variance sigma shadowing (la variance de la variable aléatoire a qui représente le shadowing comme mentionné dans l'équation 2.1). On considère $\varrho = 3.23$, $pe = 20$ dB, $d_0 = 1$ m et $f = 2.4$ GHz. Sigma de la variable aléatoire a qui représente le shadowing prend les valeurs 0, 2, 5 pour étudier l'effet de cette variation sur les performances de récupération de la matrice des distances ainsi que sur la précision de localisation.

2.3.2 Vérification de la propriété du rang faible

Pour qu'on puisse appliquer la technique de complétion de matrice, afin de récupérer la matrice MDE à partir des entrées observées uniquement, la matrice $\hat{\mathbf{X}}$ doit avoir un rang faible r . On approxime les distances inconnues à partir des distances connues parce qu'elles sont dépendantes. Pour vérifier que ce critère de rang faible de la matrice $\hat{\mathbf{X}}$ soit assuré, on applique la décomposition en valeurs singulières. La matrice $\hat{\mathbf{X}}$ de dimension $n \times n$ peut être décomposée comme suit

$$\hat{\mathbf{X}} = \mathbf{O}\mathbf{F}\mathbf{Q}^T \quad (2.42)$$

Où \mathbf{O} est une matrice unitaire de dimension $n \times n$, \mathbf{Q} est une matrice unitaire de dimension $n \times n$ et \mathbf{F} est une matrice diagonale de dimension $n \times n$. Les éléments de la diagonale de \mathbf{F} sont les valeurs singulières de $\hat{\mathbf{X}}$ qui sont organisées dans un ordre décroissant (c'est-à-dire, $\mathbf{F} = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_r, 0, \dots, 0)$). r est le rang de la matrice $\hat{\mathbf{X}}$. Si $\hat{\mathbf{X}}$ est une matrice de rang faible, ses l plus grandes valeurs singulières représentent la totalité ou presque de l'énergie totale. (c'est-à-dire, $\sum_{i=1}^l \gamma_i^2 \simeq \sum_{i=1}^n \gamma_i^2$). La métrique utilisée pour vérifier cette propriété de rang faible est la fraction de la norme nucléaire capturée par les l valeurs singulières les plus importantes. Cette fraction est définie comme suit :

$$g(l) = \frac{\sum_{i=1}^l \gamma_i^2}{\|\hat{\mathbf{X}}\|_*} = \frac{\sum_{i=1}^l \gamma_i^2}{\sum_{i=1}^n \gamma_i^2} \quad (2.43)$$

Figure 2.3 illustre la fraction de la norme nucléaire capturée par les l valeurs singulières. On constate que les 3 premières valeurs singulières capturent 90% de l'énergie de la matrice. Ce qui vérifie que $\hat{\mathbf{X}}$ est une matrice de rang faible. Dans ce cas, on peut appliquer la complétion de matrice.

2.3.3 Performances de récupération et précision de localisation

Pour évaluer les performances de différentes solutions étudiées basées sur GD et ses variantes combinées avec la trilatération, nous définissons les deux métriques suivantes :

- L'erreur de reconstruction de MDE en utilisant l'erreur quadratique moyenne

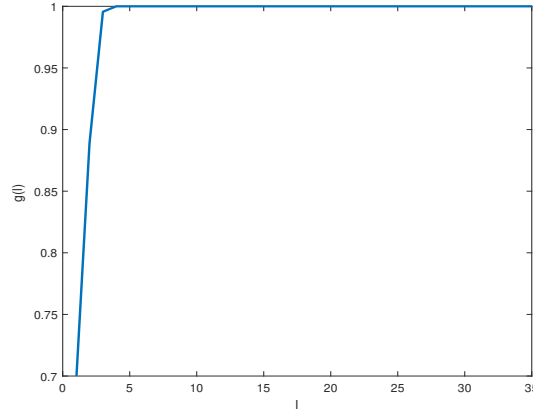


FIGURE 2.3: Fraction d'énergie capturée par les valeurs singulières.

$$MSE = \frac{\|\hat{\mathbf{X}} - \mathbf{X}_{true}\|_F^2}{\|\mathbf{X}_{true}\|_F^2} \quad (2.44)$$

— L'erreur de localisation

$$error_{loc} = \frac{\|\hat{\mathbf{C}} - \mathbf{C}\|_2}{n - m} \quad (2.45)$$

Avec $\hat{\mathbf{C}}$ est la matrice des coordonnées estimés des noeuds à localiser et \mathbf{C} sont leurs coordonnées réelles. On rappelle que α , λ , ϵ , μ , ρ , β_1 et β_2 sont des paramètres ajustés par simulation, sélectionnés pour assurer le meilleur résultat en termes d'erreur moyenne de localisation, en utilisant une base de validation. Pour trouver les meilleurs paramètres, un processus expérimental a été mené. Ainsi, plusieurs simulations ont été nécessaires pour identifier la valeur optimale de chaque paramètre. Les paramètres considérés sont présentés dans le tableau 2.2. Le reste des algorithmes à savoir GD, NAG, Adagrad, RMSProp et Adadelta partagent les mêmes valeurs optimales des paramètres.

Pour évaluer les performances de la trilatération en terme de précision de localisation, plus d'informations de distances sont fournies, nous appliquons tout d'abord la trilatération avec les distances observées uniquement. Nous pouvons facilement remarquer qu'elle engendre une précision de localisation inférieure à celle de toutes les combinaisons testées dans les environnements avec et sans bruit (Figure 2.4, Figure 2.5, Figure 2.6). La précision de localisation est ainsi bien meilleure lorsque nous utilisons une MDE complète. De plus, pour appliquer le processus de trilatération, il faut au moins trois ancres détectés. Si ce n'est pas le cas, le noeud capteur ne peut pas être localisé. Ce problème peut être résolu en utilisant une MDE contenant toutes les distances par paires. Par conséquent, la combinaison de la technique de complétion matricielle et de la trilatération est fortement recommandée.

Dans la première partie des simulations faites, nous étudions les performances, en termes de reconstruction MDE et d'erreur de localisation, de différents algorithmes cités dans un environnement sans bruit, comme illustré dans la figure 2.4. Cette figure montre l'efficacité de l'estimation de position pour 10 simulations. La figure 2.4.(a) illustre l'erreur de

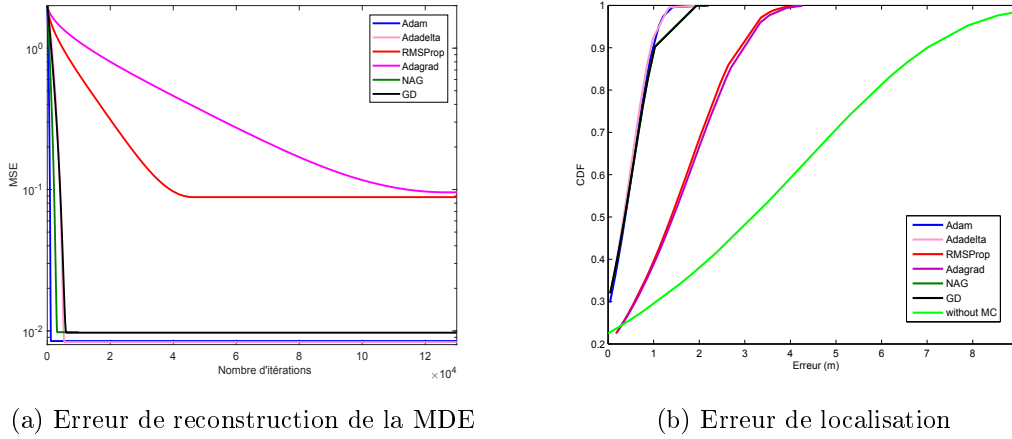


FIGURE 2.4: Performances des algorithmes avec sigma shadowing = 0.

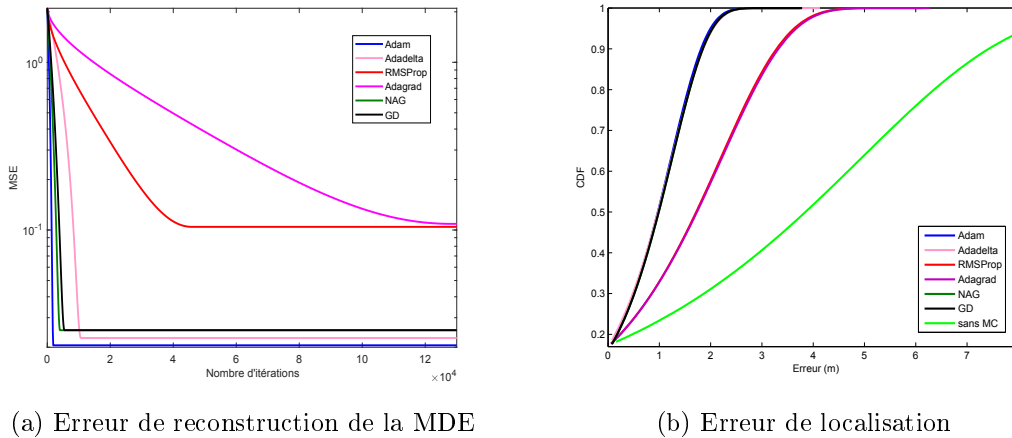


FIGURE 2.5: Performances des algorithmes avec sigma shadowing = 2.

reconstruction de la MDE 'MSE'. L'estimation d'erreur des distances connues par paires est de l'ordre de 10^{-30} . L'erreur de reconstruction varie entre 10^{-3} et 10^{-1} pour les différents algorithmes cités. Le meilleur taux de reconstruction est obtenu par Adadelta, qui produit une erreur moyenne de localisation de 0.47m. Ce résultat est proche de ceux obtenus par GD, NAG et Adam. La différence est le nombre d'itérations nécessaires pour atteindre la convergence. Adam converge en 790 itérations, ce qui correspond à 5.5% des itérations requises par Adadelta. GD converge en 5910 itérations, ce qui correspond à environ 41% des itérations requises par Adadelta.

Les performances de GD et de NAG en termes de précision de localisation sont très proches. Ainsi, leurs fonctions de distribution cumulative (CDF) sont confondues pour chaque valeur de sigma. Cependant, NAG est plus rapide, il ne nécessite que 50% du nombre d'itérations requises par le GD classique

Adagrad présente les pires performances en termes d'erreur de reconstruction de la MDE et d'erreur de localisation. Cela est dû au fait qu'il accumule les carrés des gradients dans le dénominateur. Ainsi, la somme des termes positifs continue de croître et le taux d'appren-

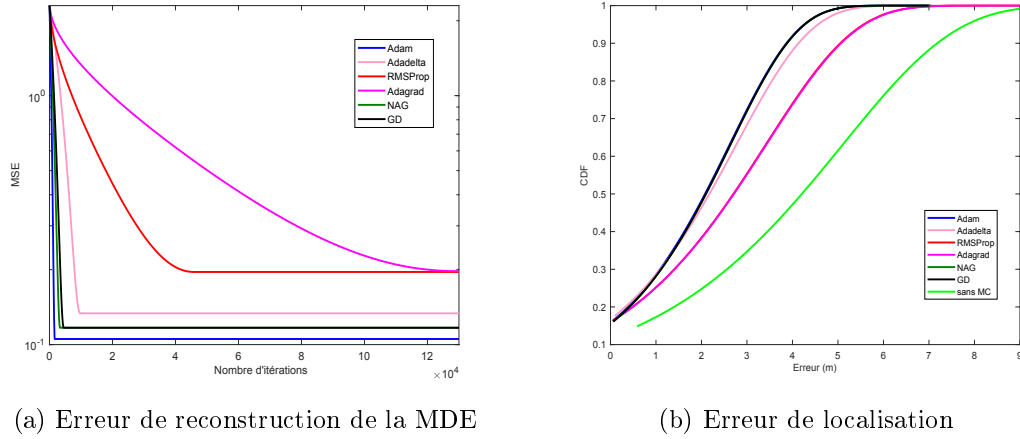


FIGURE 2.6: Performances des algorithmes avec sigma shadowing = 5.

tissage devient très faible, rendant ainsi l'algorithme incapable de garantir des mises à jour afin d'atteindre un minimum plus bas. RMSProp, Adadelta et Adam offrent une convergence rapide vers une erreur très faible.

Au lieu d'accumuler tous les carrés des gradients antérieurs, RMSProp et Adadelta utilisent une fenêtre de taille (ρ) des gradients antérieurs accumulés. RMSProp améliore un peu l'erreur de reconstruction de la MDE (Figure 2.4) et l'erreur de localisation par rapport à celles introduites par Adagrad dans un environnement sans bruit. Cependant, le résultat est toujours pire que ceux obtenus par GD et NAG. Les performances de RMSProp et d'Adagrad sont assez proches dans un environnement avec bruit. Ils convergent vers des erreurs très proches (Figure 2.4, Figure 2.5, Figure 2.6) et leurs CDF sont presque confondues.

Adadelta fonctionne un peu mieux que GD et NAG pour sigma shadowing = 0 et 2. Cependant, il lui faut 2.4 fois le nombre d'itérations requises par GD pour converger et 4.5 fois le nombre d'itérations requises par NAG pour converger (dans un environnement sans bruit). En outre, il est plus complexe que GD et NAG, ce qui rend l'algorithme inadéquat avec une localisation temps réel. Nous remarquons que adadelta est plus affecté par le bruit que les autres algorithmes. Ses performances diminuent plus rapidement que les autres. Dans un environnement intérieur sans bruit, Adadelta offre la meilleure précision de localisation. Pour un sigma shadowing égal à 2, Adam donne de meilleurs résultats. De plus, pour un sigma shadowing égal à 5, Adam, GD et NAG sont meilleurs qu'Adadelta en termes d'erreur de reconstruction de la MDE, de précision de localisation et de vitesse de convergence. L'avantage d'Adadelta est qu'il ne nécessite pas de définir un pas d'apprentissage par défaut et tous les paramètres sont définis automatiquement. Cependant, cela peut avoir un effet négatif car nous ne pouvons pas contrôler le pas d'apprentissage. Adam est introduit pour résoudre ce problème.

Adam fonctionne bien par rapport aux autres algorithmes compte tenu du compromis entre précision de localisation et temps d'exécution. Il lui faut le plus petit nombre d'itérations pour converger. Cela est dû au fait qu'il utilise un terme accélérateur en tenant compte des mises à jour antérieures. En outre, il permet d'obtenir la meilleure erreur moyenne de localisation dans un environnement en présence du bruit. Pour sigma shadowing égal à 2, son erreur moyenne de localisation est de 1.2 m et il atteint 2.6 m lorsque sigma shadowing est

TABLE 2.2: Les valeurs de différents paramètres utilisés.

	Adam	Le reste des algorithmes	
Sigma shadowing=0	α	0.5	0.5
	λ	0.9	0.9
	ϵ	0.01	0.01
	μ	–	0.9
	ρ	–	0.9
	β_1	0.001	–
	β_2	0.001	–
	Sigma shadowing=2	α	0.5
λ		0.9	0.9
ϵ		0.1	0.001
μ		–	0.35
ρ		–	0.35
β_1		0.001	–
β_2		0.001	–
Sigma shadowing=5		α	0.5
	λ	0.9	0.9
	ϵ	0.1	0.001
	μ	–	0.35
	ρ	–	0.35
	β_1	0.001	–
	β_2	0.001	–

égal à 5. Comme mentionné précédemment, ces résultats ont été réalisés sur 10 réalisations et la variance est d'environ 0.1 m pour chaque valeur de sigma shadowing. Par conséquent, Adam est l'algorithme le plus adapté aux systèmes de localisation en intérieur.

2.3.4 Etude de complexité

Expressions analytiques

Dans cette section, on vise à approximer la complexité théorique des différents algorithmes étudiés. La complexité sera évaluée en comptant le nombre de multiplications par itération et en négligeant les additions et les soustractions. Pour la multiplication de matrices, la formule de calcul utilisée est la formule classique et non la formule de Strassen [115]. Pour calculer la puissance carrée d'une matrice, la méthode binaire est considérée [116]. La complexité de la racine carrée négative d'une matrice est obtenue en calculant l'inverse de la racine carrée de cette matrice. Nous calculons la complexité de l'inverse d'une matrice selon la méthode de Gauss [115]. La complexité de la racine carrée d'une matrice est déterminée par l'algorithme de Denman-Beavers [117]. Nous définissons c_{GD} , c_{NAG} , $c_{Adagrad}$, $c_{RMSPProp}$, $c_{Adadelta}$ et c_{Adam} les complexités de calcul par itération de GD, NAG, Adagrad, RMSPProp, Adadelta et Adam, respectivement. En outre, comme défini précédemment, n est

le nombre de noeuds capteurs (anchors et noeuds inconnus).

$$c_{GD} = 4n^3 + 6n^2 \quad (2.46)$$

$$c_{NAG} = 8n^3 + 9n^2 \quad (2.47)$$

$$c_{Adagrad} = 5n^3 + 7n^2 \quad (2.48)$$

$$c_{RMSProp} = 6n^3 + 9n^2 \quad (2.49)$$

$$c_{Adadelta} = 6n^3 + 10n^2 \quad (2.50)$$

$$c_{Adam} = 6n^3 + 9n^2 \quad (2.51)$$

Analyse de la complexité

Selon les expressions analytiques déterminées dans la section précédente, il est possible de calculer numériquement la complexité des différents algorithmes, puisque la valeur de n est connue, comme indiqué dans le tableau 2.3. La complexité de chaque itération et la complexité de l'algorithme sont normalisées par rapport à la complexité de GD, afin de mettre en évidence la contribution des autres algorithmes cités par rapport à GD. L'erreur moyenne de localisation dans le tableau 2.3 correspond à la meilleure erreur moyenne enregistrée par chaque algorithme pour différentes valeurs de sigma shadowing.

Les meilleures erreurs moyennes de localisation atteintes par GD et NAG sont proches. Cependant, en termes de temps d'exécution, GD l'emporte. En termes de nombre d'itérations nécessaires pour converger, NAG converge rapidement par rapport à GD. Cependant, la complexité prise par chaque itération représente 1.9839 opérations au lieu de 1 opération enregistrée par GD. La pire erreur de localisation est introduite par Adagrad. Il est également associé à la complexité de calcul la plus importante. Le nombre d'opérations requis par itération est le meilleur comparé aux autres algorithmes utilisés (sauf GD). Cependant, le nombre d'itérations nécessaires pour converger est trop élevé, ce qui augmente considérablement la complexité de l'algorithme. La complexité de l'algorithme RMSProp est d'environ 0.4% de celle enregistrée pour Adagrad. Cependant, cette complexité reste élevée, d'où RMSProp ne peut pas être adaptée dans les larges réseaux ou dans les systèmes de localisation en temps réel.

Le nombre d'opérations requises par itération pour Adadelta et Adam est d'environ $1.5 \times c_{GD}$. Cependant, le nombre d'itérations nécessaires pour atteindre la convergence d'Adadelta est nettement supérieur au nombre requis par Adam. Et finalement, le temps d'exécution et la complexité de calcul sont plus élevés pour Adadelta que pour Adam. Adam correspond à la meilleure complexité, qui représente 0.2% de la complexité atteinte par GD. Comparé aux autres algorithmes cités, il est associé à la plus faible complexité. Il ne nécessite pas de temps de calcul élevé, il peut donc être adapté aux systèmes de localisation en temps réel. Ainsi, en considérant le compromis entre la précision de localisation et la complexité de calcul, Adam surpasse les autres algorithmes d'optimisation.

TABLE 2.3: La complexité des algorithmes normalisée par rapport à 'GD'.

	Algorithme	Erreur moyenne de localisation (m)	Nombre d'opérations par itération normalisé par le nombre d'opérations par itération de GD	Nombre d'itérations pour atteindre la convergence	Complexité de l'algorithme pour atteindre la convergence par rapport à la complexité de GD
Sigma shadow=0	Sans matrix completion	4.1	—	—	—
	GD	0.56	1	5910	1
	NAG	0.56	1.9839	3100	1.0406
	Adagrad	1.8	1.2473	120000	25.3261
	RMSProp	1.8	1.5	43750	11.1041
	Adadelta	0.47	1.5054	14150	3.6042
	Adam	0.55	1.5	790	0.2005
Sigma shadow=2	Sans matrix completion	4.7	—	—	—
	GD	1.2	1	5535	1
	NAG	1.2	1.9839	3942	1.4129
	Adagrad	2.2	1.2473	125900	28.3716
	RMSProp	2.1	1.5	45640	12.3686
	Adadelta	1.2	1.5054	10530	2.8639
	Adam	1.2	1.5	775	0.21
Sigma shadow=5	Sans matrix completion	4.9	—	—	—
	GD	2.6	1	4435	1
	NAG	2.6	1.9839	3281	1.4677
	Adagrad	3.4	1.2473	121900	34.2835
	RMSProp	3.4	1.5	43920	14.8546
	Adadelta	2.7	1.5054	9742	3.3067
	Adam	2.6	1.5	751	0.2540

2.4 Vérification de l'algorithme en utilisant une base de mesures réelles

Pour la vérification de la méthode, nous considérons une salle de classe de notre institut couvrant 88 m^2 illustrée dans la figure 2.7. La technologie utilisée est LoRa qui utilise une bande centrée autour de 868 MHz. Nous avons placé 8 noeuds capteurs capables de transmettre et de recevoir des messages. Nous recueillons des données RSSI durant un après midi. Ensuite, nous construisons la matrice de RSSI dans laquelle il manque des valeurs qui correspondent aux capteurs non détectés. Cette matrice est utilisée pour obtenir la matrice MDE partiellement connue. Nous mentionnons que la technologie utilisée et le nombre de noeuds de capteurs explorés pour vérification expérimentale sont liés à la disponibilité d'un tel équipement dans notre laboratoire. Ainsi, nous ne considérons pas les conditions optimales déterminées auparavant. Le but est de valider l'efficacité de notre méthode.

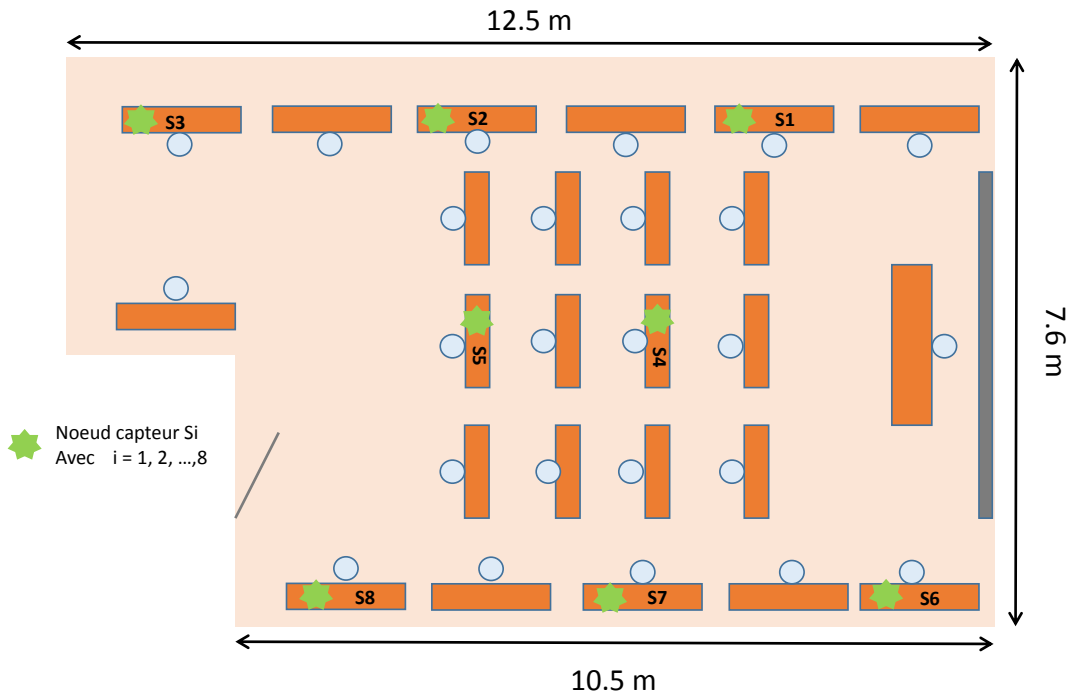


FIGURE 2.7: Configuration utilisée lors des expérimentations réelles.

Lors de l'application de notre algorithme de localisation (Adam est utilisé pour estimer la matrice MDE complète) pour localiser chaque noeud capteur, nous obtenons une erreur de localisation moyenne de 1.5 m. Cette erreur atteint 3.8 m avec la trilatération classique. Dans Figure 2.8, nous présentons la CDF correspondante à la trilatération classique (Sans MC, c'est-à-dire sans complétion de la matrice) et la CDF correspondante à notre algorithme (Avec MC, c'est-à-dire, utilisation de la complétion de matrice).

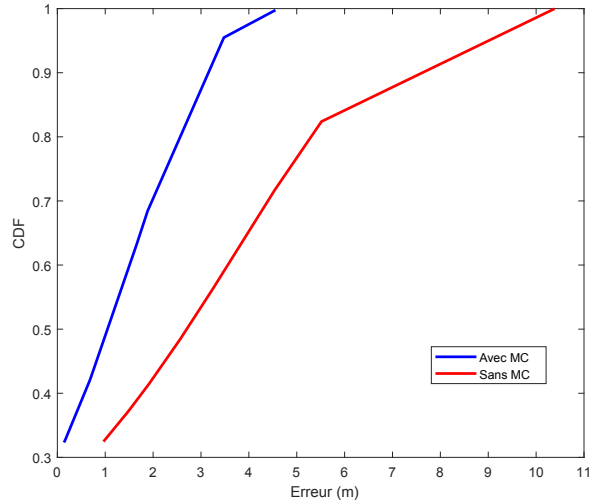


FIGURE 2.8: CDF avec des mesures réelles.

2.5 Conclusion

A la fin de ce chapitre, nous avons réussi à mettre en place un framework de localisation qui exploite la technique de complétion de matrice afin d'améliorer les performances de localisation combinée à la trilatération en fournissant plus de distances inter noeuds capteurs. Cette amélioration a été vérifiée par simulations et par mesures réelles. Le problème de complétion de matrice a été formulé comme étant un problème d'optimisation résolu par différentes variantes de GD, dont on a développé les expressions analytiques correspondantes. Le choix de différents paramètres utilisés a été justifié expérimentalement. Les performances de récupération de la matrice de distances, la précision de localisation et la complexité ont été étudiées et discutées pour chaque algorithme. Adam a finalement fourni les meilleures performances compte tenu du compromis entre précision de localisation et complexité.

Pour adopter notre approche, un capteur collecte les données RSSI provenant de tous les capteurs, applique la complétion de sa matrice de distances et se localise ou bien localise un autre noeud du réseau en utilisant la trilatération. Cet algorithme fonctionne bien et nous arrivons à améliorer considérablement les performances de localisation associées à la trilatération. Cependant, il est applicable dans des hypothèses bien déterminées :

- Nous avons travaillé dans un environnement où l'objet à localiser communique avec tous les autres objets du réseau. Ce qui n'est pas toujours garanti en IoT vu la limitation des batteries des capteurs qui limitent leurs consommation en énergie et entraîne la plupart des temps une communication entre objets et une entité centrale (station de base, passerelle, ou point d'accès).
- Tout le processus est en ligne. Lorsque la taille du réseau concerné va augmenter, cela va engendrer une augmentation de la complexité et une augmentation du temps de réponse. Ce qui peut menacer la propriété des services temps réel pour les réseaux étendus.

Donc, si les hypothèses considérées ne sont pas vérifiées, cet algorithme n'est pas optimal. Pour cela, nous allons proposer dans le chapitre suivant, des solutions de localisation où un objet communique seulement avec les entités centrales, tout en divisant le processus de localisation en une phase offline et une phase online.

Chapitre 3

Solutions de localisation pour une architecture centralisée

Dans ce chapitre, nous allons nous intéresser à la localisation quand le réseau est organisé autour d'une architecture centralisée. Dans cette dernière, tous les objets connectés communiquent avec des entités centrales (station de base, point d'accès ou passerelles). Aucune communication directe entre les objets n'est possible dans une telle organisation. La technique présentée dans le chapitre précédent devient alors non applicable. Dans le cadre de cette thèse, nous proposons d'appliquer la méthode des RSSI fingerprints présentée dans le chapitre 1. Le choix s'est porté sur le paramètre RSSI car c'est le moins coûteux en termes de mise en oeuvre. En effet, ce paramètre est mesuré en permanence par l'objet communiquant et par les stations de base afin d'estimer la qualité de la communication et de prendre des décisions sur le changement de station de base de rattachement. Comme mentionné précédemment, la localisation à base de RSSI fingerprints comporte une phase offline, durant laquelle la base d'apprentissage est construite. Puis, une phase online qui consiste à trouver la position à laquelle une nouvelle mesure a été prise. Plusieurs stratégies peuvent être adoptées :

- Evaluation de similarité : Cette méthode consiste à construire une base de fingerprints dans la phase offline et de comparer le fingerprint de test à toute la base afin de lui associer la position de celui qui est le plus proche ou une moyenne des positions les plus proches des fingerprints de la base. Dans ce cas, différentes métriques d'évaluation de similarité et de combinaison des positions des fingerprints les plus proches sont possibles. Nous allons, dans ce chapitre, étudier quelques-unes d'entre elles qui nous semblent adaptées à la nature de nos données.
- Construction d'une relation analytique, entre les valeurs des RSSI comprises dans un fingerprint et les coordonnées de la position, dans laquelle ce fingerprint a été mesuré. La difficulté d'une telle solution est qu'il est difficile de savoir, a priori, la forme de

cette relation. La démarche suivie et les résultats de localisation seront présentés dans ce chapitre.

- Construction d'un modèle basé sur les réseaux de neurones qui est capable d'établir une relation entre les valeurs des RSSI comprises dans un fingerprint et la position dans laquelle ce fingerprint a été mesuré. Des détails sur le fonctionnement des réseaux de neurones convolutifs utilisés ainsi que les résultats obtenus seront également présentés dans cette partie du manuscrit. Afin de mettre en oeuvre les stratégies présentées, un "système modèle" sera présenté sur lequel seront basées toutes les simulations menées.

3.1 Système modèle

On considère un environnement intérieur de superficie $(g \times g)$ avec M PAs placés aléatoirement ayant des positions connues. Nb positions de mesures ont été étudiées dont Nb_t positions de test, Nb_a positions d'apprentissage et Nb_v positions de validation. A chaque position, T mesures de RSSI sont prises. Un exemple d'environnement est illustré sur la Figure 3.1.

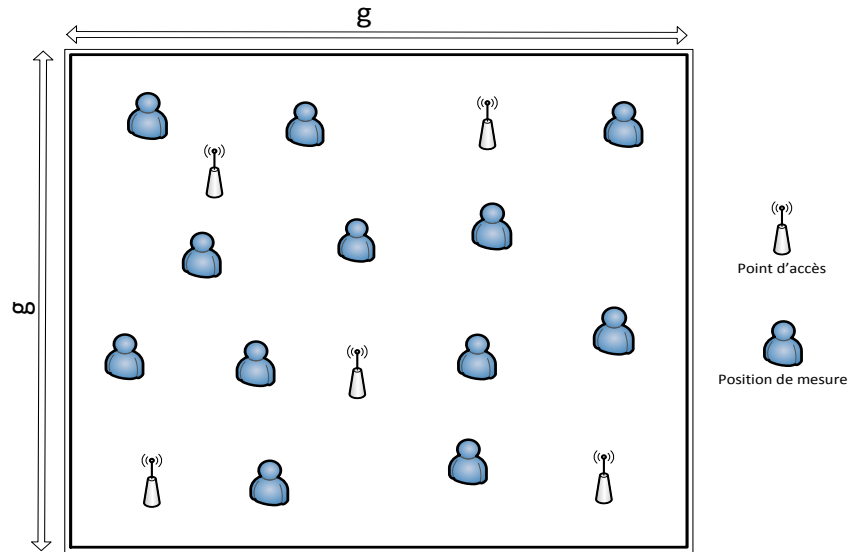


FIGURE 3.1: Exemple d'architecture avec 5 PAs et 12 positions de mesures.

3.2 Evaluation de similarité

Cette section sera consacrée à la première stratégie de localisation par fingerprints. Dans laquelle, la position du fingerprint de test est estimé en calculant la similarité avec tous les fingerprints d'apprentissage. Dans un premier temps, les différentes métriques d'évaluation de la similarité seront présentées. Ensuite, la technique d'estimation de la position sera détaillée et enfin les données utilisées et les résultats obtenus seront présentés et discutés.

3.2.1 Métriques d'évaluation

Différentes métriques d'évaluation de similarité ont été testées, elles peuvent être organisées en deux catégories. Dans ce qui suit, on suppose que le vecteur $o = (o_1, o_2, \dots, o_D)$ est le fingerprint de test contenant les RSSI reçus des D PAs. Le vecteur $o_l = (o_{l1}, o_{l2}, \dots, o_{lD})$ est le fingerprint d'apprentissage contenant les RSSI reçus des D PAs à la position l .

- Les méthodes déterministes : Dans ce cas, une distance $S(o, o_l)$ entre le fingerprint de test o et le fingerprint d'apprentissage o_l est calculée.

La distance Euclidienne : C'est la métrique la plus utilisée pour comparer les RSSI fingerprints. Elle est définie à partir de la distance de Minkowski [118] d'ordre p , avec $p = 2$.

$$S(o, o_l) = \sqrt{\sum_{j=1}^D (o_j - o_{lj})^2} \quad (3.1)$$

o_j est la valeur de RSSI reçue de la $j^{\text{ième}}$ PA au niveau d'une position de test, o_{lj} est la valeur de RSSI reçue de la $j^{\text{ième}}$ PA au niveau d'une position d'apprentissage et D est la dimension d'un fingerprint (520 dans notre cas).

La distance de Manhattan [118] : Cette distance est appelée métrique absolue. C'est un cas particulier pour $p = 1$ de la distance de Minkowski. Cette distance apporte souvent des bons résultats lorsqu'on compare des vecteurs avec une grande dimension. La distance entre fingerprints est donnée par l'expression suivante :

$$S(o, o_l) = \sum_{j=1}^D |o_j - o_{lj}| \quad (3.2)$$

- Les méthodes probabilistes : Dans ce cas, pour chaque fingerprint de test, on accorde une probabilité de similarité à chacun des fingerprints d'apprentissage.

La distance de Gauss [119] : La probabilité attribuée à un fingerprint d'apprentissage dépend des PAs détectés en accordant un coefficient de pénalité pour chaque PA non détecté. Ce coefficient permet de pénaliser les PAs non détectés. P_{exp} est la probabilité de correspondance assignée pour un fingerprint d'apprentissage o_l en considérant les M PAs détectés ou non détectés.

$$P_{exp} = \sqrt[M]{\prod_{j=1}^M e^{-\left(\frac{o_j - o_{lj}}{\sigma}\right)^2}} \quad (3.3)$$

Avec M est le nombre des PAs détectés ou non détectés dans le fingerprint de test et celui d'apprentissage et σ est un paramètre réglable. Pour l'ensemble de T PAs non détectés dans le fingerprint de test et non pas dans le fingerprint d'apprentissage, une pénalité P_{pen_T} est calculée. Cette pénalité augmente quand le RSSI du PA détecté uniquement dans le fingerprint d'apprentissage est plus important que la valeur de RSSI minimale détectée dans le fingerprint d'apprentissage $\min(o_l)$.

$$P_{pen_T} = \prod_{j=1}^T e^{-\left(\frac{o_j - \min(o_l)}{\sigma}\right)^2} \quad (3.4)$$

Un autre terme de pénalité est aussi utilisé pour les N PAs détectés uniquement dans le fingerprint d'apprentissage. Ce terme augmente quand la différence entre le RSSI

de ce PA est plus important que la valeur minimale du fingerprint de test $\min(o)$.

$$P_{pen_N} = \prod_{j=1}^N e^{-\left(\frac{o_j - \min(o)}{\sigma}\right)^2} \quad (3.5)$$

Les deux termes de pénalité sont ensuite combinés pour former une seule pénalité donnée par l'expression suivante :

$$P_{pen} = \sqrt[N+T]{P_{pen_T} \times P_{pen_N}} \quad (3.6)$$

La probabilité de similarité P calculée pour chaque fingerprint d'apprentissage est une fonction de P_{exp} et P_{pen} . On note que $N + M + T = D$.

$$P(o, o_l) = \sqrt{P_{pen} \times P_{exp}} \quad (3.7)$$

La méthode de Kernel Gaussien [120] : Une probabilité est assignée pour chaque fingerprint d'apprentissage en utilisant Kernel Gaussien. C'est une méthode particulière de la méthode de Gauss ou on n'accorde pas un coefficient de pénalité pour les PAs non détectés.

$$P_{Gauss}(o, o_l) = \prod_{j=1}^D \frac{1}{2\pi\sigma^2} e^{\left(\frac{-(o_j - o_{lj})^2}{2\sigma^2}\right)} \quad (3.8)$$

3.2.2 Estimation de la position

En ce qui concerne l'estimation de la position $c_{est} = (lat_{est}, lon_{est})$ du fingerprint de test, les positions $c_l = (lat_l, lon_l)$ des fingerprints d'apprentissage les plus similaires ont été combinées de trois façons différentes :

— Moyenne simple

Cette méthode consiste à moyennner les coordonnées (latitude et longitude) des positions d'apprentissage correspondant aux fingerprints les plus similaires au fingerprint de test (les K plus proches voisins).

$$lat_{est} = \frac{\sum_k lat_k}{K} \quad (3.9)$$

$$lon_{est} = \frac{\sum_k lon_k}{K} \quad (3.10)$$

— Moyenne pondérée

Une approche plus sophistiquée consiste à associer un poids $w(k)$ pour les positions d'apprentissage correspondant aux K plus proches fingerprints d'apprentissage. Ce poids permet d'augmenter l'impact des positions les plus proches. Dans le cas des méthodes déterministes, $w(k)$ est donné par :

$$w(k) = \frac{1}{S(o, o_l)} \quad (3.11)$$

Dans le cas des méthodes probabilistes, ce poids est égal à la probabilité de similarité.

$$w(k) = P(o, o_l) \quad (3.12)$$

Enfin, la latitude et la longitude du point de test sont calculées comme suit :

$$lat_{est} = \frac{\sum_k lat_k \times w(k)}{\sum_k w(k)} \quad (3.13)$$

$$lon_{est} = \frac{\sum_k lon_k \times w(k)}{\sum_k w(k)} \quad (3.14)$$

— Estimateur de Nadaraya Watson Kernel [121]

C'est une méthode d'estimation probabiliste qui utilise comme estimateur de position l'espérance conditionnelle $E(c_{est}|o)$. Cet estimateur minimise l'erreur moyenne de localisation. Pour cela, on utilise l'estimateur de Nadaraya Watson Kernel où $G(o - o_k)$ est une fonction de Kernel. Intuitivement, cet estimateur est la moyenne pondérée normalisée des coordonnées des plus proches fingerprints d'apprentissage. Les coordonnées c_{est} du fingerprint o sont calculées en utilisant les coordonnées des K fingerprints d'apprentissage.

$$E(c_{est}|o) \approx \frac{\sum_{k=1}^K c_k G(o - o_k)}{\sum_{k=1}^K G(o - o_k)} \quad (3.15)$$

Où

$$G(o - o_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} e^{(-\frac{1}{2}(o - o_k)^T \cdot (o - o_k))} \quad (3.16)$$

3.2.3 Présentation des données

Pour tester nos approches développées dans ce chapitre, nous avons utilisé une base publique de RSSI fingerprints UJIndoorLoc [122]. UJIndoorLoc est une base de données introduite pour combler le gap de l'inexistence d'une base de données commune. C'est la plus grande base accessible au public, construite pour des fins de comparaison. Les données ont été collectées par une équipe de chercheurs de l'Université de Jaume I, Espagne en 2013. Cette base de données contient des mesures de RSSI reçues de 520 PAs couvrant l'environnement étudié prises à des positions connues appelées positions de référence ou positions d'apprentissage. Elle couvre 108703 m² comprenant 3 bâtiments multi étages. 19936 fingerprints d'apprentissage et 1110 fingerprints de validation ont été collectées avec 25 équipements différents utilisés par 20 utilisateurs. Un utilisateur peut utiliser différents appareils lors de la collecte de données. Les fingerprints d'apprentissage ont été recueillis à 933 positions de référence. Les mesures de validation ont été capturées 4 mois après l'apprentissage. Chaque fingerprint contient 520 RSSI de différents PAs et 9 étiquettes détaillées dans le tableau 3.1.

3.2.4 Présentation et discussion des résultats et évaluation de performances

Toutes les combinaisons des métriques d'évaluation de similarité/techniques d'estimation de la position ont été testées. Les erreurs de localisation loc_{erreur} correspondantes à toutes les combinaisons sont présentés ci-après. Cette erreur est donnée par

$$loc_{erreur} = \sqrt{(lat_{est} - lat_{obs})^2 + (lon_{est} - lon_{obs})^2} \quad (3.17)$$

lon_{est} et lat_{est} sont les coordonnées estimés et lon_{obs} et lat_{obs} sont les coordonnées observées.

TABLE 3.1: Description des champs d'un fingerprint de la base UJIndoorLoc.

Nom du Champs	Description
Mesures de RSSI	Ces valeurs correspondent à la mesure de la puissance de signal reçue de chaque PA à une position donnée. Ces valeurs vont de -104 dBm à 0 dBm. La valeur de RSSI associée aux PAs non détectés est définie par 100 . Dans notre travail, nous avons attribué -110 dBm à ces PAs afin d'avoir un signal qui n'affecte pas le processus de calcul.
Latitude et longitude	Les Coordonnées associées aux mesures de RSSI.
ID de l'étage	L'étage contenant la position. Cette valeur varie de 0 à 4 .
ID du bâtiment	Comme 3 bâtiments sont couverts, ce champs peut avoir la valeur $0, 1$ et 2 .
ID de l'espace	Cette valeur est utilisée pour identifier l'espace (bureau, laboratoire, etc.).
Position relative	On peut avoir 1 ou 2 . 1 signifie à l'intérieur et 2 signifie dans le couloir devant la porte d'un espace donné.
ID de l'utilisateur	Ce champs varie de 0 à 18 . Il définit l'utilisateur qui a collecté ces mesures.
ID du téléphone	Ce champ indique le modèle et la version de l'appareil utilisé pour faire les mesures.
Horlodotage	L'heure à laquelle le fingerprint a été pris.

Pour valider notre système, nous avons utilisé la base UJIndoorLoc présenté au début de ce chapitre. Seules les mesures dérivées du bâtiment 0 ont été utilisées pour évaluer les algorithmes détaillés ci-dessus et réduire la complexité. Pour identifier le paramètre K (le nombre des plus proches voisins) qui garantit la meilleure erreur moyenne de localisation, on a fait varier K de 1 à 20 . La position est estimée lorsque le bâtiment et l'étage sont correctement estimés. On a enregistré les meilleures performances lorsque K atteint 5 . Dans ce qui suit, nous considérons les 5 plus proches voisins.

Nous avons également vérifié la meilleure valeur de K qui assure le meilleur taux de réussite dans l'estimation de l'étage. Nous remarquons que le meilleur taux est obtenu avec 1 plus proche voisin pour toutes les distances introduites. Le meilleur taux de réussite est obtenu en utilisant la distance Euclidienne ($4, 4\%$ d'erreur) et le pire est fourni par la distance de Gauss (23% d'erreur).

Nous rappelons que σ est un paramètre ajustable. Sa valeur varie de 1 à 20 pour la distance de Gauss et la méthode de Kernel Gaussien. Nous avons gardé la valeur qui assure le meilleur résultat en termes d'erreur moyenne de localisation et d'erreur d'estimation de l'étage. Dans ce qui suit, la valeur de σ utilisée par la distance de Gauss vaut 5 et celle

utilisée par Kernel Gaussien est 10.

TABLE 3.2: L'erreur moyenne de localisation en mètres correspondante à chaque combinaison.

	Moyenne simple	Moyenne pondérée	Estimateur de Nadaraya
Distance Euclidienne	5.51 m	5.5 m	5.59 m
Distance de Manhattan	5.38 m	5.25 m	5.4 m
Distance de Gauss	7.63 m	7.7 m	7.81 m
Méthode de Kernel	5.51 m	5.9 m	5.59 m

Selon les résultats préliminaires, la moyenne pondérée s'est avérée être la solution d'estimation la plus performante avec les métriques déterministes et la moyenne simple est meilleure avec les métriques probabilistes (meilleures méthodes d'estimation de position qui ont été prouvées expérimentalement illustrées dans tableau 3.2). Ces choix ont été retenus pour le reste du document.

Le meilleur résultat en termes d'erreur moyenne est obtenu en utilisant la distance de Manhattan comme métrique d'évaluation de similarité et la moyenne pondérée des K plus proches voisins pour estimer les coordonnées réelles du mobile. La distance Euclidienne est la métrique la plus utilisée pour comparer les RSSI fingerprints. Pour la distance de Manhattan, elle apporte des bons résultats lorsqu'on compare des vecteurs avec une grande dimension. Dans ce cas, elle assure des performances meilleures que celles obtenues par la distance Euclidienne. L'erreur moyenne enregistrée par la distance de Manhattan est légèrement inférieure à celle introduite par la distance Euclidienne quel que soit la méthode d'estimation de position utilisée. Ces métriques d'évaluation de similarité appelées déterministes ont une faible complexité de calcul. Les résultats obtenus par Kernel sont très proches de ceux obtenus par la distance Euclidienne. Cela est expliqué par le fait que Kernel combiné avec KNN présente un cas particulier de la distance Euclidienne lorsque σ se rapproche de 0. La méthode de Kernel accorde une probabilité assez faible aux PAs non détectés dans le fingerprint de test ou bien celui d'apprentissage. Kernel pénalise donc les mesures non importantes. Cette probabilité est inférieure au coefficient de pénalité calculé par la distance de Gauss. Ce qui explique que l'erreur moyenne introduite par cette dernière est plus importante que celle obtenue par Kernel. Elle est de l'ordre de 7.64 m. L'utilisation de la méthode de Gauss en attribuant un coefficient de pénalité aux PAs non détectés peut être une bonne solution si on compare cette métrique à d'autres, qui utilisent seulement les PAs détectés dans la phase de test et celle d'apprentissage. En effet, si on n'utilise que les PAs détectés, les fingerprints d'apprentissage ayant moins de PAs détectés vont être plus favorisés par rapport aux autres fingerprints d'apprentissage qui ont plus de PAs détectés. Dans ce cas, l'utilisation d'un coefficient de pénalité peut améliorer l'erreur moyenne de localisation.

L'erreur de localisation peut atteindre 25 m. L'écart important entre l'erreur moyenne de localisation et l'erreur maximale introduite peut être expliqué par plusieurs facteurs :

1. Les mesures d'apprentissage sont prises en 259 positions dans un bâtiment avec quatre étages couvrant une superficie supérieure à 108703 m² (soit 420 m² / point de mesure).

2. Les mesures de validation ont été prises 4 mois après celle d'apprentissage (des changements dans l'environnement peuvent se produire).
3. Les mesures d'apprentissage et celles de validation ne sont pas prises avec le même appareil.
4. Faible couverture de quelques endroits.

Pour évaluer les performances de notre meilleure combinaison retenue, nous l'avons comparé avec d'autres algorithmes existants, qui utilisent des mesures réelles de RSSI. Puisque chaque équipe de recherche présente ses résultats dans son propre environnement intérieur, nous proposons d'utiliser un paramètre d'évaluation de performance relatif à la superficie de l'environnement intérieur. Nous avons appelé ce paramètre 'Densité d'erreur'. C'est le rapport entre l'erreur moyenne et la superficie de l'environnement. Le tableau 3.3 est un tableau comparatif qui illustre la densité d'erreur, la superficie de l'environnement intérieur et l'erreur moyenne correspondante à notre combinaison gagnante (la distance de Manhattan et la moyenne pondérée) et à 3 autres approches existantes. A partir de ce tableau, nous pouvons remarquer que si on se base sur l'erreur moyenne c'est la méthode SeamLoc (qui sera précisée dans le paragraphe suivant) qui l'emporte cependant, ce résultat n'est pas significatif comparé à la superficie de l'environnement étudié. En se basant sur la densité de l'erreur de localisation, nous remarquons que le meilleur résultat obtenu est enregistré par notre approche lorsque l'erreur de positionnement utilisée est relative à la superficie de l'environnement.

TABLE 3.3: Tableau comparatif des approches utilisant des mesures réelles de RSSI.

	Erreur moyenne (m)	Superficie de l'environnement intérieur (m²)	Densité d'erreur (m⁻¹)
Approche proposée	5.25	12428	4.2×10^{-4}
SeamLoc	1.81	834.08	2.2×10^{-3}
PCA interpolation [123]	2.91	432.96	6.7×10^{-3}
Exponential Kernel [124]	5.4	–	–

Vu le grand nombre des points d'accès non détectés contenu dans un fingerprint donné (en moyenne 18 PAs parmi 520 sont détectés) et pour alléger le calcul et la complexité de nos algorithmes, on a proposé de diminuer le nombre des PAs tout en gardant l'information utile sans affecter la précision de localisation. On a recours à une sélection des PAs pour enlever l'information redondante et inutile. La sélection avec Gram Schmidt n'a pas répondu à nos besoins marquant une dégradation de l'erreur moyenne. Alors que celle à base de l'entropie de Shannon a réussi à maintenir l'erreur moyenne introduite sans sélection des PAs. L'idée d'utiliser l'entropie comme critère de sélection revient au fait que, pour chaque étage, il y a un nombre important de PAs qui ont une entropie nulle ou très faible. Ce qui indique qu'elle n'apporte pas d'informations concernant la localisation. Par exemple, pour le 3^{eme} étage du bâtiment 0, 39 PAs parmi 201 PAs détectés ont une entropie nulle (voir Figure 3.2)

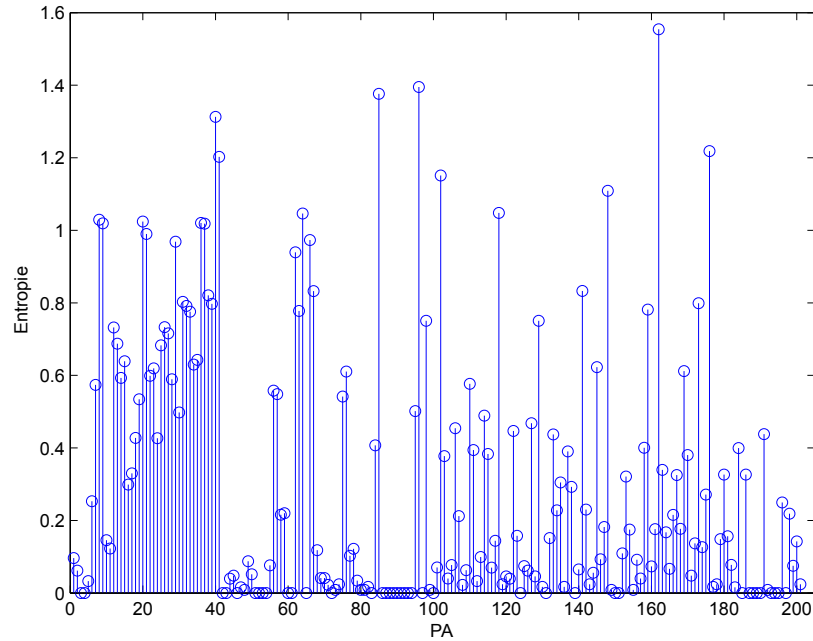


FIGURE 3.2: La valeur d'entropie correspondant à chaque PA détecté au niveau de l'étage 3 du bâtiment 0.

3.3 Sélection de PAs

Afin de diminuer la haute dimension de la base UJIndoorLoc, certaines approches introduisent un processus de sélection tout en gardant l'information utile au lieu d'utiliser toute la base. Les auteurs de [125] ont mis en oeuvre une approche de localisation appelée 'SeamLoc'. Cette approche nécessite un nombre réduit de points d'apprentissage, ce qui est dû à un processus d'interpolation. L'interpolation est basée sur la construction de fonctions de vraisemblance, décrivant la probabilité qu'un utilisateur se trouve à proximité d'un point d'apprentissage. Les fonctions de probabilité sont obtenues en utilisant une formulation Bayes Naïve.

Au lieu de sélectionner les PAs, ce qui évite une complexité fastidieuse, l'algorithme proposé dans [126] transforme les valeurs de RSSI en composants principaux (PC). Cet algorithme réorganise les informations afin de réduire la dimension en combinant les PAs. Ce processus évite les calculs inutiles tout en garantissant la précision. Une approche probabiliste est adoptée pour produire une estimation de position à partir de PC dérivés en utilisant une approximation distribuée Gaussienne. Une probabilité a posteriori est calculée selon la règle de Bayes. La position estimée est celle qui correspond à la probabilité la plus élevée ou la moyenne pondérée des positions de référence.

Contrairement à ce qui a été fait, nous proposons de faire une sélection des PAs dont l'information des RSSI est pertinente pour la localisation. La principale question pour faire

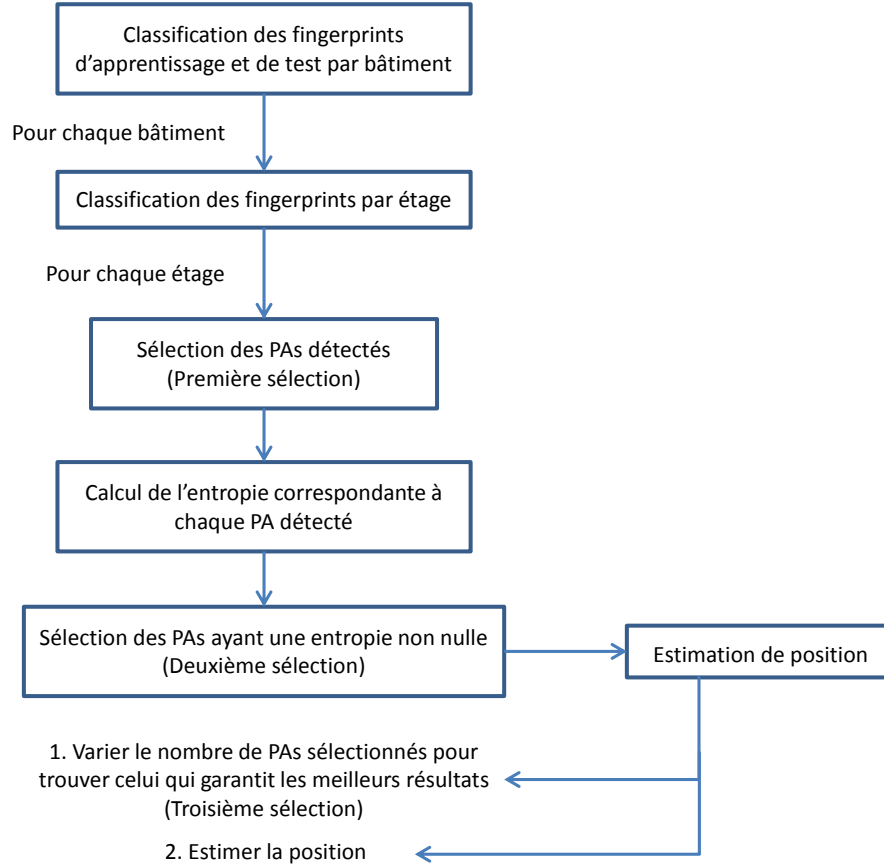


FIGURE 3.3: Les étapes de l'algorithme de localisation assurant une sélection des PAs.

la sélection est le critère choisi. Dans le cadre de cette thèse, nous nous sommes concentré sur l'utilisation de l'entropie de l'information contenue dans les mesures des RSSI de chaque PA. Si l'entropie d'un PA est importante, cela veut dire que les RSSI mesurés sont pertinents pour la localisation. Pour la base de données de chaque étage, les étapes suivantes ont été appliquées. Le processus est décrit sur la Figure 3.3 :

Première étape (Première sélection) : Sélection des PAs détectés au moins une fois pendant la phase d'apprentissage ou bien de validation.

Deuxième étape : Pour quantifier la diversité d'informations contenues dans les mesures RSSI reçues de différents PAs concernant la position, une métrique est proposée. Elle se base sur l'entropie de Shannon. L'entropie S_i calculée pour un PA est donnée par l'équation suivante :

$$S_i = - \sum_{b=1}^B P_{i,b} \log(P_{i,b}) \quad (3.18)$$

Où B est le nombre d'intervalles englobant les valeurs de RSSI de chaque PA. Cette valeur

est choisie expérimentalement. $P_{i,b}$ est la probabilité qu'une valeur de RSSI reçue soit dans un intervalle donné pour les positions d'apprentissage.

$$P_{i,b} = \frac{1}{N} \sum_{j=1}^N p_{i,b,j} \quad (3.19)$$

Où N est le nombre des positions d'apprentissage.

Troisième étape (Deuxième sélection) : Une valeur d'entropie est associée pour chaque PA. Plus cette valeur est importante, plus l'information apportée par ce PA concernant la position est importante. A ce niveau, on sélectionne les PAs ayant une entropie non nulle.

Quatrième étape : On estime la position de test. D'abord, on évalue la similarité entre le fingerprint de test et tous les fingerprints d'apprentissage. Pour cela, une probabilité de similarité est accordée pour chaque position d'apprentissage. Cette probabilité est calculée en utilisant la méthode de kernel décrite dans la section précédente.

Ensuite, les coordonnées de l'utilisateur lon_{est} et lat_{est} sont estimées comme une combinaison pondérée des L positions d'apprentissage en utilisant la moyenne pondérée des K plus proches voisins.

3.3.1 Présentation et discussion des résultats

Pour calculer l'entropie de Shannon, nous devons prédéfinir le nombre d'intervalles qui assure les meilleurs résultats. Pour choisir la meilleure valeur, on a fait varier le nombre d'intervalles de 10 à 100. On a constaté que ce paramètre n'affecte pas la précision de localisation. Dans la suite, nous considérons 100 intervalles.

Nous avons 13 zones de test qui correspondent aux étages des 3 bâtiments de l'Université. La superficie de chacun est supérieure à 10000 m². 520 PAs ont été employés. Etant donné le grand nombre de PAs utilisés, un processus de sélection de PAs est appliqué. Il comporte trois phases.

- Pour chaque étage, il existe un nombre important de PAs non détectés. Nous considérons que les PAs détectés au moins une fois dans la phase d'apprentissage ou de test. Le nombre de PAs sélectionnés est mentionné dans le tableau 3.3. Il passe de 520 pour atteindre des valeurs beaucoup plus basses. Il varie entre 96 PAs pour l'étage 0 du bâtiment 2 et 203 PAs détectés dans le deuxième étage du bâtiment 1.
- Pour assurer la deuxième sélection, on applique l'entropie de Shannon. Pour chaque PA détecté, nous calculons l'entropie associée. Nous éliminons les PAs ayant une entropie nulle. Le nombre de PAs sélectionnés est également mentionné dans le tableau 3.3. Il varie de 88 PAs à 190 PAs.
- Lors de l'estimation de position, le nombre de PAs varie de 10 au nombre de PAs sélectionnés au cours de la deuxième sélection, afin de conserver le nombre qui garantit les meilleurs résultats en terme d'erreur moyenne. Les résultats obtenus sont présentés dans la troisième colonne du tableau 3.3.

Pour estimer la position d'un utilisateur, une combinaison de positions d'apprentissage est effectuée. Nous avons considéré différentes combinaisons en faisant varier le nombre

TABLE 3.4: Le nombre de PAs sélectionné correspondant à chaque phase de sélection.

	Première sélection	Deuxième sélection	Troisième sélection
Bâtiment0_Etage0	135	119	90
Bâtiment0_Etage1	167	133	100
Bâtiment0_Etage2	198	168	100
Bâtiment0_Etage3	201	162	60
Bâtiment1_Etage0	130	121	80
Bâtiment1_Etage1	177	168	168
Bâtiment1_Etage2	203	190	130
Bâtiment1_Etage3	164	146	110
Bâtiment2_Etage0	96	88	50
Bâtiment2_Etage1	171	161	100
Bâtiment2_Etage2	158	149	70
Bâtiment2_Etage3	183	178	140
Bâtiment2_Etage4	136	119	40

de positions à combiner. Nous avons constaté que si nous combinons toutes les positions d'apprentissage pondérés avec leurs probabilités de similarité, nous obtenons les meilleurs résultats.

En utilisant seulement les PAs sélectionnés, nous avons tracé les CDFs correspondantes à chaque étage des 3 bâtiments (Figure 3.4, Figure 3.5 et Figure 3.6). Pour bâtiment 0, l'erreur moyenne atteinte est d'environ 5 m pour ses différents étages. Cette erreur varie entre 8.56 m (étage 3) et 18.96 m (étage 0). Pour bâtiment 2, l'erreur moyenne obtenue est de 10.93 m. Pour certains étages, l'erreur de localisation peut atteindre des valeurs élevées, mais avec une faible probabilité. On constate aussi un écart important entre l'erreur médiane et l'erreur moyenne.

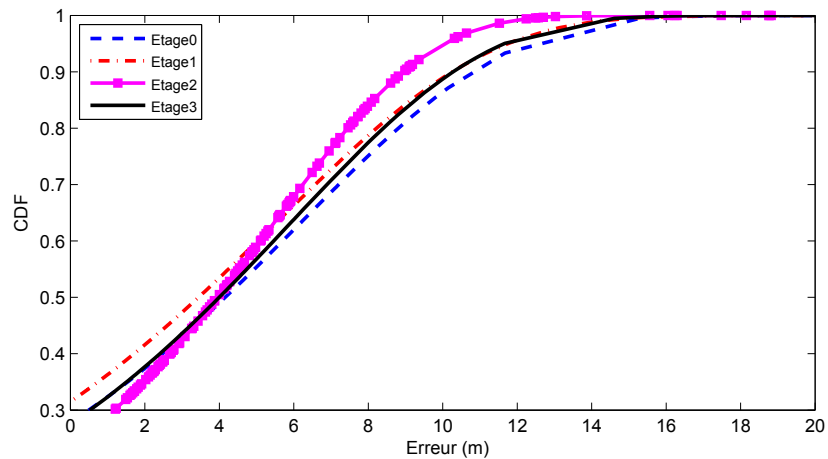


FIGURE 3.4: CDFs de différents étages du bâtiment 0.

Afin de comparer notre approche par rapport aux autres approches développées dans

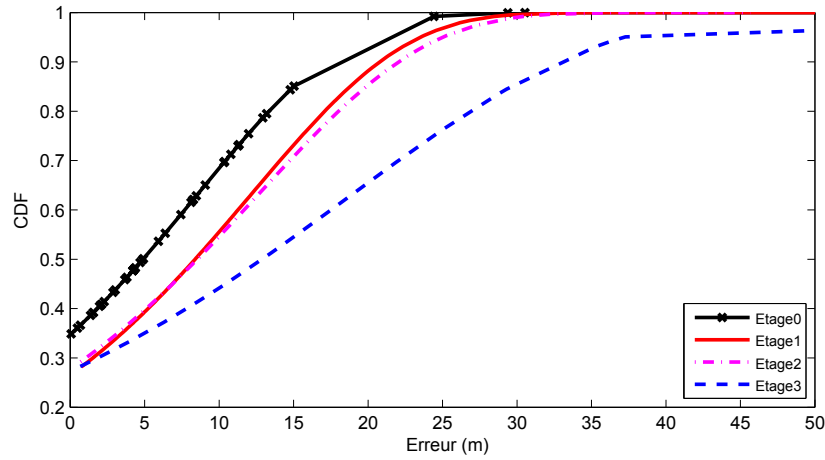


FIGURE 3.5: CDFs de différents étages du bâtiment 1.

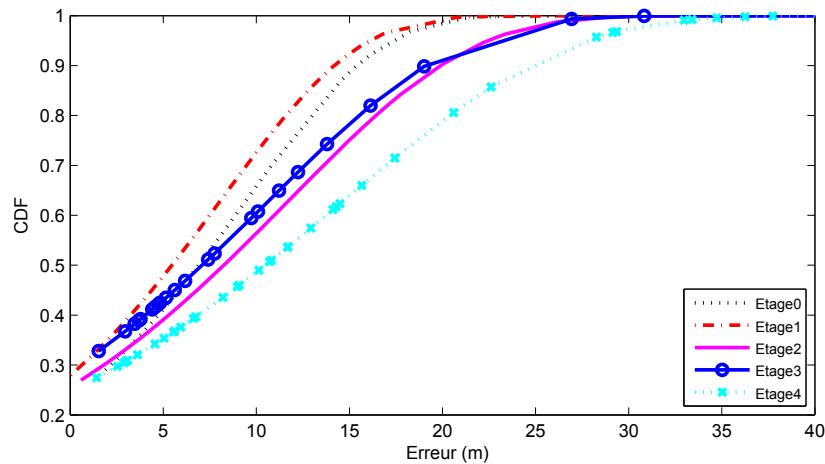


FIGURE 3.6: CDFs de différents étages du bâtiment 2.

la littérature exploitant la base UJIndoorLoc et estimant la position dans un étage donné, comme paramètres de comparaison, nous utilisons l'erreur moyenne et l'erreur médiane correspondant à chaque environnement de test (Etage). Le tableau 3.5 présente l'erreur moyenne et le tableau 3.6 présente l'erreur médiane, associées à notre approche et deux autres approches développées par les constructeurs de la base UJIndoorLoc (KNN [127] et la méthode probabiliste existante [127]).

Nous constatons facilement que l'approche probabiliste existante est associée à la pire précision de localisation comparée à KNN et notre approche en termes d'erreur moyenne et d'erreur médiane. KNN appliqué à toutes les mesures RSSI donne de bons résultats. Ceci est dû au fait que KNN offre de bonnes performances avec les bases de données volumineuses comme UJIndoorLoc. Les résultats enregistrés par KNN et notre approche sont trop proches. Concernant l'erreur moyenne, nous pouvons observer une légère dégradation pour notre algorithme par rapport à KNN pour les trois bâtiments : 0.36 m pour bâtiment 0, 0.17 m pour bâtiment 1 et 0.26 m pour bâtiment 2. Concernant l'erreur médiane, les meilleurs

TABLE 3.5: L'erreur moyenne en mètres pour chaque étage.

	Méthode proposée	KNN	Méthode probabiliste
Bâtiment0_Etage0	6.18	5.7	8.09
Bâtiment0_Etage1	5.52	5.1	6.4
Bâtiment0_Etage2	5.46	5.4	5.7
Bâtiment0_Etage3	5.91	5.6	6.79
Bâtiment0	5.76	5.4	6.74
Bâtiment1_Etage0	18.96	18.46	18.83
Bâtiment1_Etage1	12.17	12.4	22.7
Bâtiment1_Etage2	12.66	12.3	14.45
Bâtiment1_Etage3	8.56	8.49	8.81
Bâtiment1	13.08	12.91	16.19
Bâtiment2_Etage0	10.75	10.69	13.87
Bâtiment2_Etage1	8.12	8.4	59.74
Bâtiment2_Etage2	11.8	11.9	12.39
Bâtiment2_Etage3	9.25	8.5	9.54
Bâtiment2_Etage4	14.77	13.9	21
Bâtiment2	10.93	10.67	13.3

TABLE 3.6: L'erreur médiane en mètres pour chaque étage.

	Méthode proposée	KNN	Méthode probabiliste
Bâtiment0_Etage0	4.16	4.75	6.28
Bâtiment0_Etage1	3.44	3.35	4.77
Bâtiment0_Etage2	3.93	4.45	4.76
Bâtiment0_Etage3	3.96	4.03	3.99
Bâtiment0	3.88	4.14	4.95
Bâtiment1_Etage0	12.85	14.69	14.96
Bâtiment1_Etage1	8.4	6.95	15.18
Bâtiment1_Etage2	8.55	9.07	10.33
Bâtiment1_Etage3	4.9	5.17	7.65
Bâtiment1	8.67	8.97	12.03
Bâtiment2_Etage0	7.08	5.92	9.83
Bâtiment2_Etage1	5.48	6.83	7.83
Bâtiment2_Etage2	8.25	6.56	8.37
Bâtiment2_Etage3	6.89	6.45	6.69
Bâtiment2_Etage4	10.47	8.43	14.72
Bâtiment2	7.63	6.83	9.48

résultats sont obtenus en appliquant notre méthode dans 7 étages. Nous avons donc réussi à maintenir la précision de positionnement enregistrée par KNN en utilisant un nombre réduit

de PAs (le nombre de PAs utilisés est divisé par 5 comparé au nombre total de PAs) sans perte importante d'information. La légère dégradation peut être due aux erreurs de mesures.

Vu le manque d'informations fournies par les constructeurs de la base UJIndoorLoc (les paramètres de propagation utilisés, les positions des points d'accès, les positions des points d'apprentissage, etc), nous ne pouvons plus l'utiliser pour évaluer des systèmes avancés. Pour cela, les prochaines approches proposées vont être évaluées en utilisant le système de propagation avec des paramètres réels fixés selon les spécificités de notre laboratoire, approuvés expérimentalement auparavant.

3.4 Localisation à base d'un système analytique

En se basant sur différentes combinaisons possibles des méthodes classiques déterministes ou probabilistes, les solutions proposées dans les sections précédentes sont prometteuses garantissant une bonne précision de localisation. Mais, pour évaluer un système de localisation, il y a un compromis entre précision de localisation et complexité. Ces systèmes sont associés à des complexités de calcul importantes ou nécessitent des capacités de calcul et de stockage importantes. En effet, il est intuitif de penser que plus la base de fingerprints d'apprentissage est importante plus le résultat de localisation est précis. Cependant, plus le nombre de fingerprints d'apprentissage augmente, plus le temps passé à comparer le fingerprint de test à cette base augmente. Ce qui introduit une latence dans l'estimation de la position qui n'est pas toujours compatible avec les applications envisagées. Pour remédier à cet inconvénient, l'estimation d'une relation analytique entre les RSSI et les coordonnées peut être une solution. Dans cette section, nous développons une nouvelle méthode de localisation en intérieur à l'aide des RSSI fingerprints, basée sur un processus d'optimisation, en essayant d'estimer une matrice reliant les coordonnées de la position de l'utilisateur à localiser aux valeurs mesurées de RSSI, reçues par cet utilisateur, de différents PAs déployés dans l'environnement intérieur. L'objectif est de mettre en place un système linéaire pour établir une fonction analytique entre les coordonnées et les valeurs de RSSI. Ceci va permettre de minimiser la complexité de calcul en ligne en appliquant directement un modèle optimisé selon les fingerprints d'apprentissage.

3.4.1 Développement de Taylor pour une localisation en intérieur

Dans l'environnement intérieur étudié où les utilisateurs cherchent à se localiser, on considère N points d'apprentissage identifiés par leurs positions (x_i, y_i, z_i) . Pour chaque position, nous associons un vecteur de fingerprints formé de K RSSI moyens reçus des différents PAs déployés. Les positions des PAs ainsi que leurs puissances émises sont inconnues. Les transmissions peuvent être en Line Of Sight (LOS) ou Non Line Of Sight (NLOS). On note par $A_{j, j \in [1, K]}$ le j^{ieme} PA ayant comme coordonnées inconnues (x'_j, y'_j, z'_j) . Nous omettons les termes liés au fast fading compte tenu du fait que nous travaillons avec la moyenne des puissances reçues sur une période du temps. La puissance moyenne reçue au niveau du point d'apprentissage M_i venant du PA A_j notée $p_{i,j}$ est essentiellement liée à l'affaiblissement du parcours. Elle peut être exprimée comme suit :

$$p_{i,j} = \frac{\alpha_{i,j} p e_j}{d_{i,j}^{\gamma_{i,j}}} \quad (3.20)$$

$p_{i,j}$ est la puissance émise par le PA A_j , $\alpha_{i,j}$ est un terme constant, $\gamma_{i,j}$ est l'exposant de l'affaiblissement par rapport à la distance, $d_{i,j}$ est la distance entre le PA A_j et le point d'apprentissage M_i . En développant l'équation 3.20 en tenant compte des coordonnées des PAs et des positions d'apprentissage, on obtient :

$$p_{i,j} = \frac{\alpha_{i,j} p e_j}{\sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2 + (z_i - z'_j)^2}^{\gamma_{i,j}}} \quad (3.21)$$

La puissance moyenne reçue peut donc être exprimée comme une fonction de (x_i, y_i, z_i) :

$$p_{i,j} = f(x_i, y_i, z_i) \quad (3.22)$$

Nous proposons d'introduire un développement de Taylor de $f(x_i, y_i, z_i)$. Nous présentons un développement limité au deuxième ordre dans l'équation 3.23 qui introduit un point central ayant comme coordonnées (a, b, c) . Idéalement, ce point doit être proche de tous les points M_i . Dans ce cas, la puissance moyenne reçue par chaque M_i exprimée par $f(x_i, y_i, z_i)$ est donnée en considérant $f(a, b, c)$ et toutes les dérivées partielles calculées en ce point. Pour être conforme à ces hypothèses, nous devons envisager un espace intérieur pas trop large.

$$f(x_i, y_i, z_i) = \left\{ \begin{array}{l} f(a, b, c) \\ + (x_i - a) \frac{\delta f}{\delta x_i}(a, b, c) \\ + (y_i - b) \frac{\delta f}{\delta y_i}(a, b, c) \\ + (z_i - c) \frac{\delta f}{\delta z_i}(a, b, c) \\ + \frac{1}{2!} ((x_i - a)^2 \frac{\delta^2 f}{\delta x_i^2}(a, b, c)) \\ + \frac{1}{2!} ((y_i - b)^2 \frac{\delta^2 f}{\delta y_i^2}(a, b, c)) \\ + \frac{1}{2!} ((z_i - c)^2 \frac{\delta^2 f}{\delta z_i^2}(a, b, c)) \\ + \frac{1}{2!} (2(x_i - a)(y_i - b) \frac{\delta^2 f}{\delta x_i \delta y_i}(a, b, c)) \\ + \frac{1}{2!} (2(x_i - a)(z_i - c) \frac{\delta^2 f}{\delta x_i \delta z_i}(a, b, c)) \\ + \frac{1}{2!} (2(y_i - b)(z_i - c) \frac{\delta^2 f}{\delta y_i \delta z_i}(a, b, c)) \end{array} \right. \quad (3.23)$$

En considérant le développement de Taylor dans l'équation 3.23, nous remarquons que la fonction $f(x_i, y_i, z_i)$ peut être exprimée comme une combinaison linéaire de $(1, x_i, y_i, z_i, x_i y_i, x_i z_i, y_i z_i, x_i^2, y_i^2, z_i^2)$. Dans ce travail, nous limitons le développement de Taylor au second ordre comme présenté dans équation 3.23.

Pour chaque point d'apprentissage, nous introduisons un vecteur étendu de coordonnées

C_i de dimension $(L \times 1)$ (dans notre cas, $L = 10$) défini comme suit :

$$C_i = \begin{pmatrix} 1 \\ x_i \\ y_i \\ z_i \\ x_i^2 \\ y_i^2 \\ z_i^2 \\ x_i y_i \\ x_i z_i \\ y_i z_i \end{pmatrix} \quad (3.24)$$

Pour chaque point d'apprentissage, on introduit un vecteur de puissances moyennes reçues des K PAs. Ce vecteur est de dimension $(K \times 1)$:

$$p_i = \begin{pmatrix} p_{i,1} \\ p_{i,2} \\ \vdots \\ p_{i,K} \end{pmatrix} \quad (3.25)$$

L'objectif de cet algorithme est d'identifier une matrice \mathbf{M} de dimension $(K \times L)$ telle que :

$$P_i = \mathbf{M}C_i \quad (3.26)$$

Pour identifier la matrice \mathbf{M} , nous introduisons le critère de l'erreur quadratique moyenne $J(\mathbf{M})$

$$J(\mathbf{M}) = \sum_{i=1}^N \|P_i - \mathbf{M}C_i\|^2 \quad (3.27)$$

Ce critère va être minimisé par rapport à \mathbf{M} : $\min_{\mathbf{M}} J(\mathbf{M})$. La solution optimale est obtenue en annulant la dérivée de ce critère par rapport à M :

$$\hat{\mathbf{M}} / \frac{\delta J(\hat{\mathbf{M}})}{\delta \mathbf{M}} = 0 \quad (3.28)$$

Ce qui mène à obtenir la solution exprimée par l'équation 3.29 :

$$\hat{\mathbf{M}} = \left(\sum_{i=1}^N P_i C_i^T \right) \left(\sum_{i=1}^N C_i C_i^T \right)^{-1} \quad (3.29)$$

Une fois nous avons estimé $\hat{\mathbf{M}}$, notre système est prêt à estimer les coordonnées inconnues (x_u, y_u, z_u) d'un utilisateur U à localiser. Le vecteur étendu des coordonnées de U est obtenu en utilisant le pseudo-inverse de la matrice $\hat{\mathbf{M}}$ comme suit :

$$\hat{C}_u = (\hat{\mathbf{M}}^T \hat{\mathbf{M}})^{-1} \hat{\mathbf{M}}^T P_u \quad (3.30)$$

TABLE 3.7: Description de l'algorithme de localisation avec un développement de Taylor de l'ordre 2.

Initialisation
Placer N points d'apprentissage ou de référence dans l'environnement intérieur et générer leurs vecteurs de coordonnées associés C_i tel que $i \in [1, N]$
Mesurer les puissances moyennes reçues de différents PAs en ces points afin de construire les vecteurs p_i avec $i \in [1, N]$
Estimation de $\hat{\mathbf{M}}$
Calculer : $\hat{\mathbf{M}} = (\sum_{i=1}^N P_i C_i^T) ((\sum_{i=1}^N C_i C_i^T)^{-1})$
Calculer : $(\hat{\mathbf{M}}^T \hat{\mathbf{M}})^{-1}$
Localisation d'un utilisateur U
Mesurer les puissances moyennes reçues :
$p_u = (p_{u,1}, p_{u,2}, \dots, p_{u,K})^T$
Estimer le vecteur étendu des coordonnées de U :
$\hat{C}_u = (\hat{\mathbf{M}}^T \hat{\mathbf{M}})^{-1} \hat{\mathbf{M}}^T P_u$
Retourner les coordonnées estimées de U ($\hat{x}_u, \hat{y}_u, \hat{z}_u$) qui correspondent à la 2 ^{ième} , 3 ^{ième} et 4 ^{ième} composantes de \hat{C}_u

L'algorithme complet est illustré dans tableau 3.7. Il fonctionne sans avoir besoin de connaître ni les positions des PAs, ni leurs puissances émises, ni les exposants de l'environnement $\gamma_{i,j}$.

Deux inversions de matrices sont requises par notre algorithme. La première inversion concerne la matrice $\sum_{i=1}^N C_i C_i^T$ de dimension $(L \times L)$. Le rang de cette matrice est généralement égal à L . Il peut être contrôlé en offline en faisant un choix bien étudié des points d'apprentissage M_i . La deuxième inversion présentée par l'équation 3.30 concerne la matrice $\hat{\mathbf{M}}^T \hat{\mathbf{M}}$ de dimension $L \times L$. Le rang de cette matrice ne peut pas être facilement contrôlé et il dépend fortement de la matrice $\sum_{i=1}^N P_i C_i^T$, elle-même fortement impactée par les positions des PAs. Un test de rang doit être proposé afin d'éviter les cas où le déterminant de la matrice $\hat{\mathbf{M}}^T \hat{\mathbf{M}}$ est proche de 0, lorsque les PAs ne sont pas bien distribués dans l'espace intérieur étudié.

3.4.2 Evaluation des performances et validation du système

Pour évaluer les performances de notre système et étudier l'impact de variation de plusieurs paramètres sur ces performances, nous avons mis en place plusieurs simulations. Puisque nous travaillons dans un plan 2D, le vecteur étendu de coordonnées va être réduit à un vecteur de dimension (6×1) : $C_i = (1, x_i, y_i, x_i y_i, x_i^2, y_i^2)^T$. Les $K = 6$ PAs sont placés aléatoirement dans l'espace. Pour les besoins de simulation, nous avons choisi d'utiliser le modèle de propagation ITU [128] qui exprime la perte de propagation moyenne comme étant :

$$Loss = 20 \log(f) + 10 \gamma \log(d) \quad (3.31)$$

Où la fréquence f est exprimée en MHz, la distance d en mètres et γ est l'exposant de

la distance (dans cette section, nous considérons que $\gamma_{i,j} = \gamma \forall i, j$). Dans nos simulations, nous avons considéré la technologie WiFi avec des PAs qui opèrent à une fréquence $f = 2.4$ GHz. Nous obtenons $20\log(f) = 67.6$ correspondant à $\gamma_{i,j} = 10^{-6.76}$ dans équation 3.21 $\forall i, j$. Comme puissance émise des PAs, nous avons choisi 20 dBm.

Etude de l'impact de la variation de $\gamma_{i,j}$ sur les performances de localisation

Nous avons considéré 26 points d'apprentissage placés dans l'environnement étudié avec un pas de 25 m entre 2 points adjacents. Figure 3.7 illustre ce scénario typique avec les PAs (en diamants noirs) et les points d'apprentissage placés sur une grille (en croix bleues).

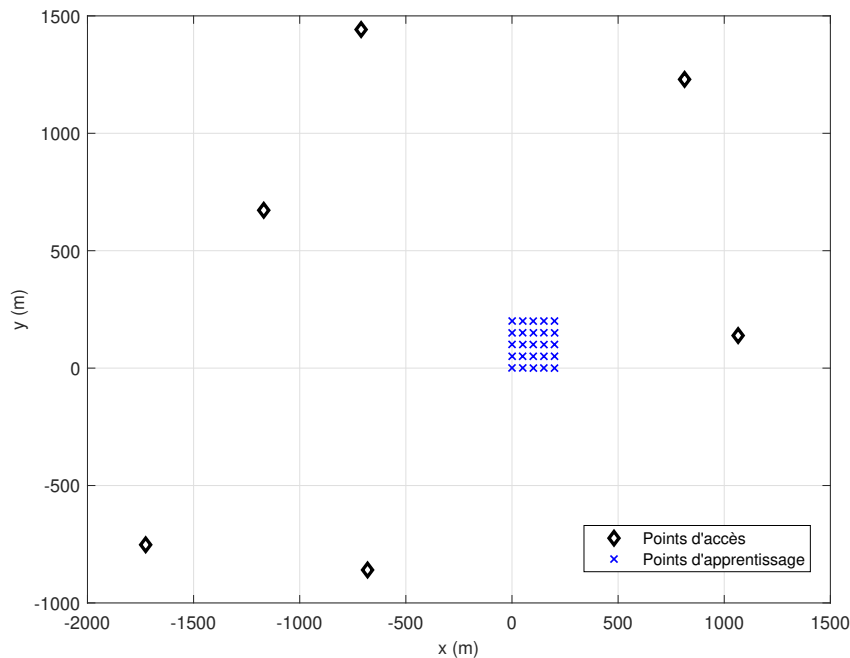


FIGURE 3.7: Scénario d'une configuration en intérieur avec 6 PAs et 25 points d'apprentissage.

Nous avons généré 1000 scénarios en changeant l'emplacement des PAs et pour chaque cas, nous avons généré aléatoirement 500 points de test afin d'évaluer les performances de l'algorithme proposé. Les Figures 3.8, 3.9 et 3.10 présentent les fonctions de densité cumulative des erreurs de localisation pour $\gamma = 2$, $\gamma = 3$ et $\gamma = 4$. Une augmentation de la valeur de γ entraîne une diminution de la puissance moyenne reçue associée à une dégradation de la précision de localisation. Comme l'algorithme est basé sur les mesures des puissances reçues, nous avons ajouté une incertitude de mesure qui varie de 1% à 5% pour ces mesures. Pour une erreur de mesure de l'ordre de 1%, nous avons 90% de chance que l'erreur moyenne de localisation soit inférieure à 2.5 m pour $\gamma = 2$. Cette valeur atteint 4 m pour $\gamma = 4$.

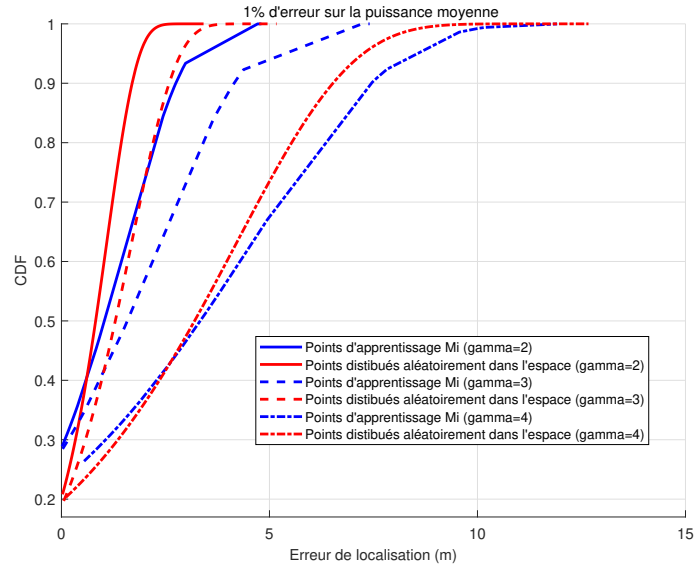


FIGURE 3.8: CDF de l'erreur de localisation avec 1% d'erreur sur les puissances reçues pour 25 points d'apprentissage.

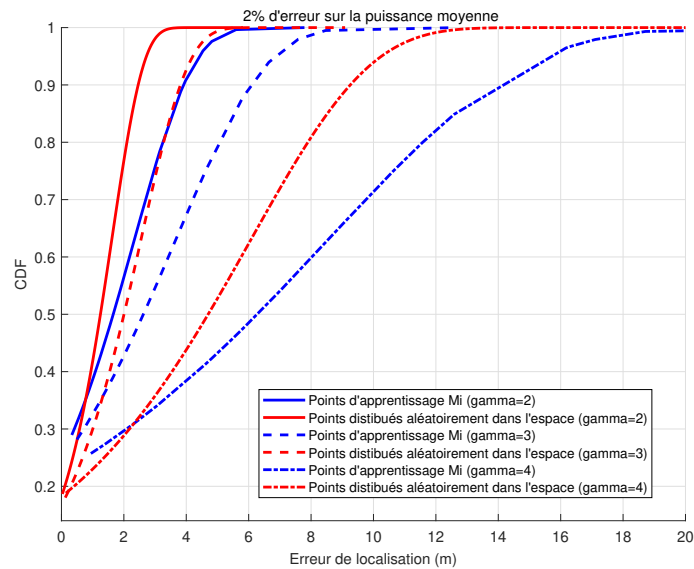


FIGURE 3.9: CDF de l'erreur de localisation avec 2% d'erreur sur les puissances reçues pour 25 points d'apprentissage.

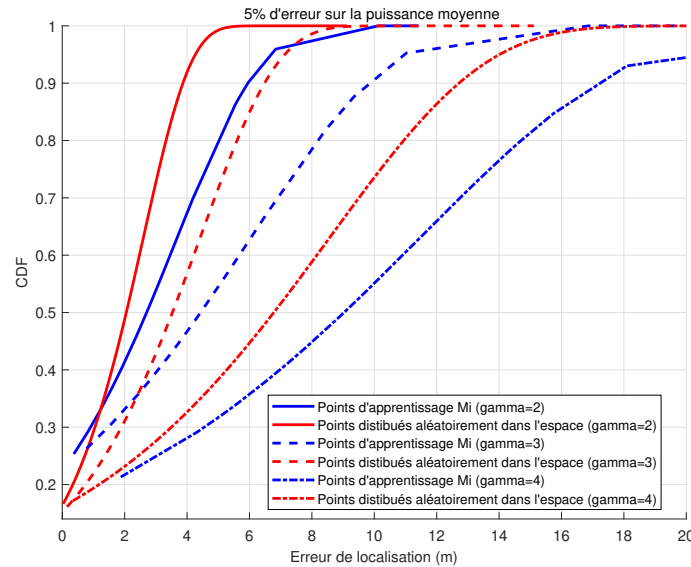


FIGURE 3.10: CDF de l'erreur de localisation avec 5% d'erreur sur les puissances reçues pour 25 points d'apprentissage.

Variation du nombre de points d'apprentissage

Après la variation de γ , nous avons fait varier le nombre de points d'apprentissage placés dans l'espace étudié. Au lieu de travailler avec 25 points, nous avons placé seulement 9 points d'apprentissage avec un pas de 100 m entre les points adjacents. Le tableau 3.8 présente l'erreur de localisation moyenne en mètres pour 90% des cas.

Nous pouvons facilement constater, à partir des figures, que les performances de localisation diminuent lorsque le nombre de points d'apprentissage est petit. Avec 5% d'erreur sur les mesures de puissances reçues et pour $\gamma = 4$ en exploitant 9 points d'apprentissage, nous obtenons une erreur moyenne de localisation inférieure ou égale à 30 m pour 90% des cas. Alors que cette erreur est inférieure à 19.5 m lorsque nous travaillons avec 25 points d'apprentissage. Ceci confirme bien que la diminution du nombre de points d'apprentissage affecte considérablement la précision de localisation.

L'impact de la distance moyenne entre les PAs et le centre de l'espace étudié

Pour cela, nous avons testé une seule configuration où les PAs sont placés à 1000 m et 2000 m du centre. Nous avons placé 4 PAs très proches les uns des autres. Nous avons remarqué que lorsque les PAs sont éloignés entre eux, les puissances moyennes reçues sont mieux distribuées et les performances de l'algorithme sont meilleures. Dans ce scénario et avec $\gamma = 3$ (Figure 3.11), lorsque la distance est égale à 2000 m, nous avons une erreur de localisation inférieure à 6 m dans 90% des cas. Alors que l'erreur de localisation peut atteindre 22 m lorsque la distance est égale à 1000 m.

TABLE 3.8: Erreur de localisation en mètres pour 90% de points avec $N = 25$ et $N = 9$ pour différentes valeurs d'erreurs sur les puissances moyennes reçues.

Nombre de points d'apprentissage	Valeur de γ	Incertitude de mesure de puissance moyenne	Erreur de localisation (m)
N = 25	$\gamma = 2$	1%	2.5
		2%	4
		5%	5.5
	$\gamma = 3$	1%	3
		2%	6
		5%	7.5
	$\gamma = 4$	1%	7.5
		2%	8
		5%	19.5
N = 9	$\gamma = 2$	1%	3.5
		2%	4
		5%	10
	$\gamma = 3$	1%	6
		2%	7
		5%	15
	$\gamma = 4$	1%	9
		2%	10.5
		5%	30

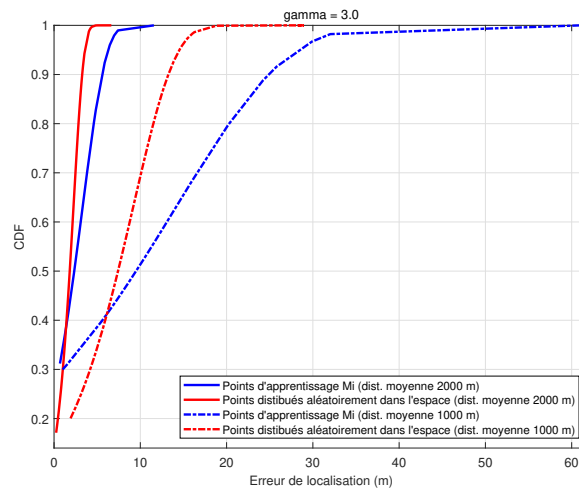


FIGURE 3.11: CDF de l'erreur de localisation pour $\gamma = 3$ avec 25 points d'apprentissage et des PAs placés à 1000 m ou 2000 m du centre de l'espace intérieur.

3.5 Localisation avec un modèle à base des réseaux de neurones convolutifs CNN

Comme nous avons vu dans la section précédente, la localisation peut être assurée en mettant en place un système analytique reliant le vecteur de puissances moyennes reçues en un point à ses coordonnées, afin de minimiser la complexité du processus de prédiction en ligne de localisation. Cependant, ce système n'est pas toujours accessible vu les contraintes et les hypothèses à respecter afin de le mettre en place. Nous avons donc proposé une approche prédictive basée sur une phase d'apprentissage. Pour cela, nous proposons d'utiliser les CNNs en formulant le problème de localisation comme un problème de classification d'images radios de 3 dimensions. L'idée consiste à obtenir, à la sortie du réseau, la zone contenant l'utilisateur à localiser. Nous commençons par introduire notre système modèle. Puis, nous expliquons en détails les différentes étapes de notre algorithme et enfin nous évaluerons ses performances.

3.5.1 Application du Deep CNN dans la localisation en intérieur

Notre système modèle comporte deux phases (Figure 3.12) : une phase offline comprenant la collecte et le prétraitement des données à utiliser comme entrées pour le modèle CNN et l'apprentissage de ce dernier. Puis, une phase online introduite pour trouver la position de chaque noeud de capteur dans l'environnement étudié en utilisant le modèle après apprentissage. Le local étudié a tout d'abord, été divisé en différentes partitions nommées "classes", comme le montre la figure 3.13. Chaque zone a été étiquetée classe q avec $q \in 1, 2, \dots, Q$ et Q est le nombre de classes. Nous mentionnons que la superficie de chaque classe est un choix, basé sur la précision de localisation requise par l'application et la disponibilité des ressources informatiques. Dans ce qui suit, chaque étape du processus de localisation va être détaillée.

3.5.2 Les données

- **Acquisition et normalisation des données** : Au niveau de chaque point d'apprentissage, T mesures de RSSI consécutives, reçues de M PAs sont prises. N bases de RSSI, appelées réalisations sont construites pour chaque point d'apprentissage, comme illustré dans Figure 3.14 ; où N est le nombre de réalisations, T le nombre de mesures RSSI reçues de chaque PA, et M est le nombre de PA. Par conséquent, chaque réalisation présente les valeurs de RSSI reçues de différents PAs aux instants T . On précise que N et T sont ajustés expérimentalement.

Le facteur le plus important en deep learning est la quantité de données disponibles pour l'apprentissage ainsi que la pertinence de ces données. En outre, il est généralement nécessaire de normaliser les données, en particulier lors de l'utilisation de méthodes d'optimisation par gradient, afin d'accélérer le processus d'apprentissage et de minimiser les risques de divergence de l'algorithme. Les données présentées à l'entrée du CNN peuvent inclure des informations supplémentaires qui peuvent être pertinentes pour la localisation. Dans le cadre de cette thèse, le choix s'est porté sur le Kurtosis.

- **Calcul de Kurtosis** : Afin d'améliorer l'efficacité de notre approche, nous avons

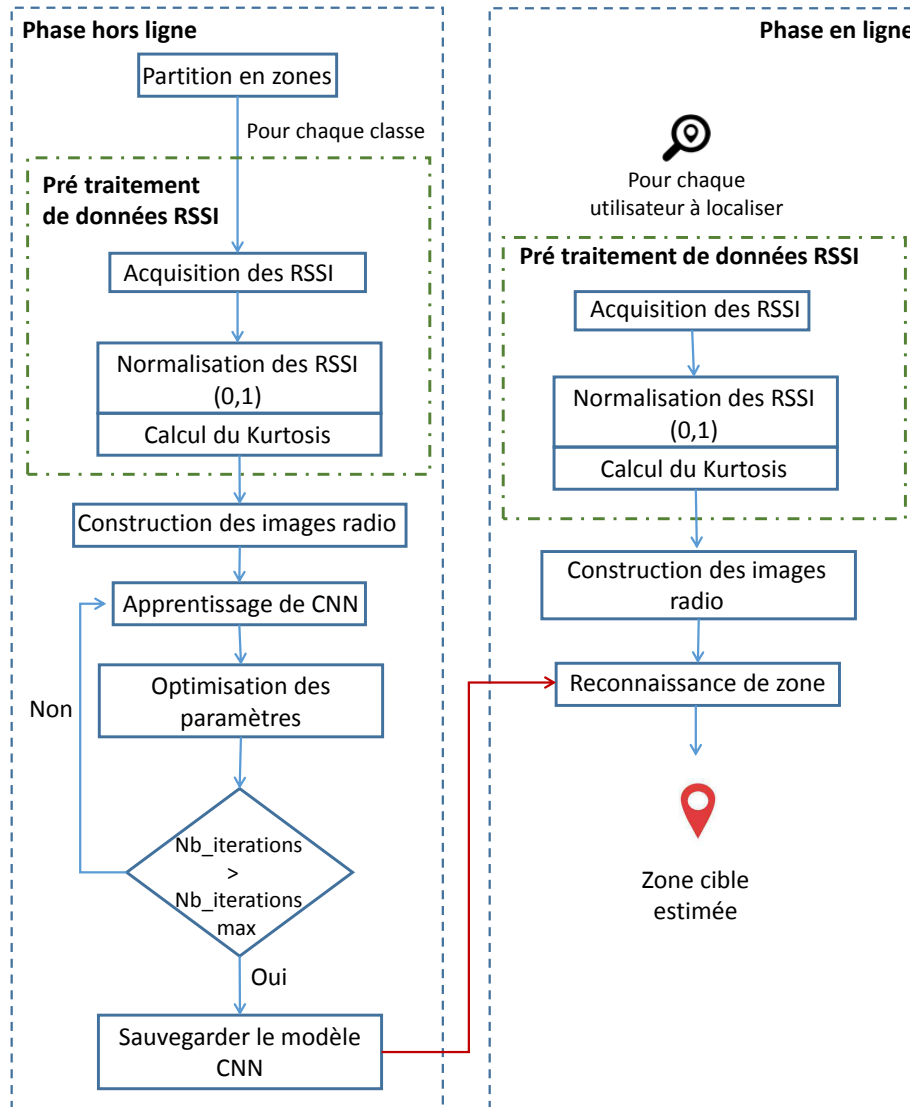


FIGURE 3.12: Les différentes étapes de localisation à base du CNNLocWK.

utilisé le kurtosis comme deuxième paramètre à utiliser avec les RSSI. Cela permet de fournir de nouvelles informations à notre réseau. En utilisant ces dernières, nous cherchons à fournir des informations statistiques calculées à partir des valeurs de RSSI pouvant présenter des informations utiles (informations globales de l'image d'entrée). Le kurtosis étant le moment d'ordre 4, il apporte des informations non linéaires, qui peuvent être utiles et non redondantes, puisque les opérations assurées par les couches de convolution du réseau de neurones sont des opérations linéaires. Il a été défini par Karl Pearson comme le moment d'ordre 4 [129]. R_{mt} est la valeur de RSSI reçue du PA m à l'instant t , où $m = 1, 2, \dots, M$ et $t = 1, 2, \dots, T$. Pour un noeud de capteur spécifique, le kurtosis est calculé comme suit :

3.5 Localisation avec un modèle à base des réseaux de neurones convolutifs CNN 87

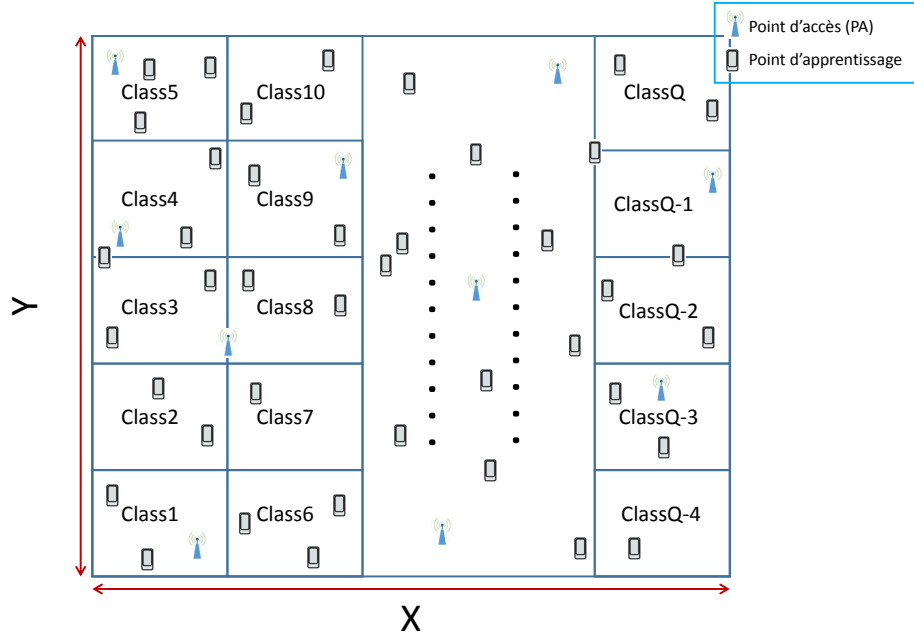


FIGURE 3.13: Partition de l'environnement intérieur en zones.

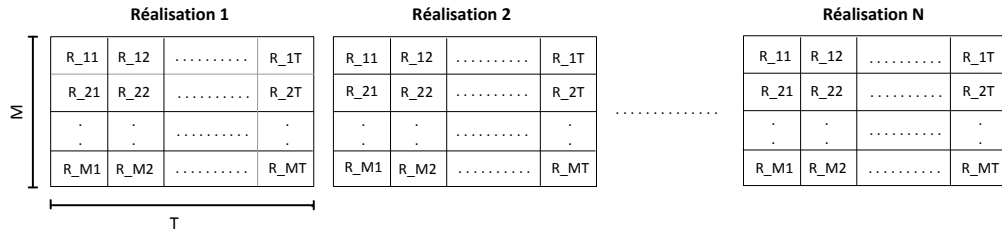


FIGURE 3.14: La structure des bases RSSI pour chaque point d'apprentissage.

$$kur_{mk} = \frac{1}{T} \times \sum_{t=1}^T \left(\frac{R_{mt} - \mu_k}{\sigma_k} \right)^4, \quad (3.32)$$

Où

$$\mu_k = \frac{\sum_{m=1}^M R_{mk}}{M}, \quad (3.33)$$

Et

$$\sigma_k = \frac{\sum_{m=1}^M R_{mk}^2}{M}, \quad (3.34)$$

Après avoir collecté les valeurs de RSSI et calculé les valeurs de kurtosis correspondant à chaque base de RSSI, des tenseurs radio sont formés. Comme première dimension

du tenseur, on a le nombre de PAs M . La deuxième dimension, on a le paramètre T qui représente le nombre de valeurs de RSSI mesurées à partir de M PAs. Et la troisième dimension, c'est le nombre de paramètres exploités dans la localisation qui est 2 (RSSI et kurtosis). Ainsi, la taille de chaque réalisation est devenue $(M \times T \times 2)$ (Figure 3.15). Les images radio construites à partir des tenseurs radio contiennent donc les valeurs de RSSI dans le premier plan et les valeurs de kurtosis dans le deuxième plan. Elles devaient être classifiées et organisées, de sorte que chaque image soit étiquetée q , $q = 1, 2, \dots, Q$. Ensuite, N réalisations de chaque noeud du capteur doit appartenir à une classe associée. Les images sont organisées dans Q dossiers étiquetés $class1, class2, \dots, classQ$, chacun contenant les images radio appropriées.

Pour trouver la position d'un capteur, après l'acquisition des valeurs RSSI et le pré-traitement des données, une image radio est construite avec la même dimension et la même structure que celles utilisées en apprentissage. Cette image est transmise au modèle après apprentissage afin de prédire la zone contenant le noeud de capteur. Pour cela, des probabilités sont attribuées à chaque classe en vérifiant bien que la somme de ces probabilités est égale à 1. La classe prédite est celle qui correspond à la probabilité la plus élevée.

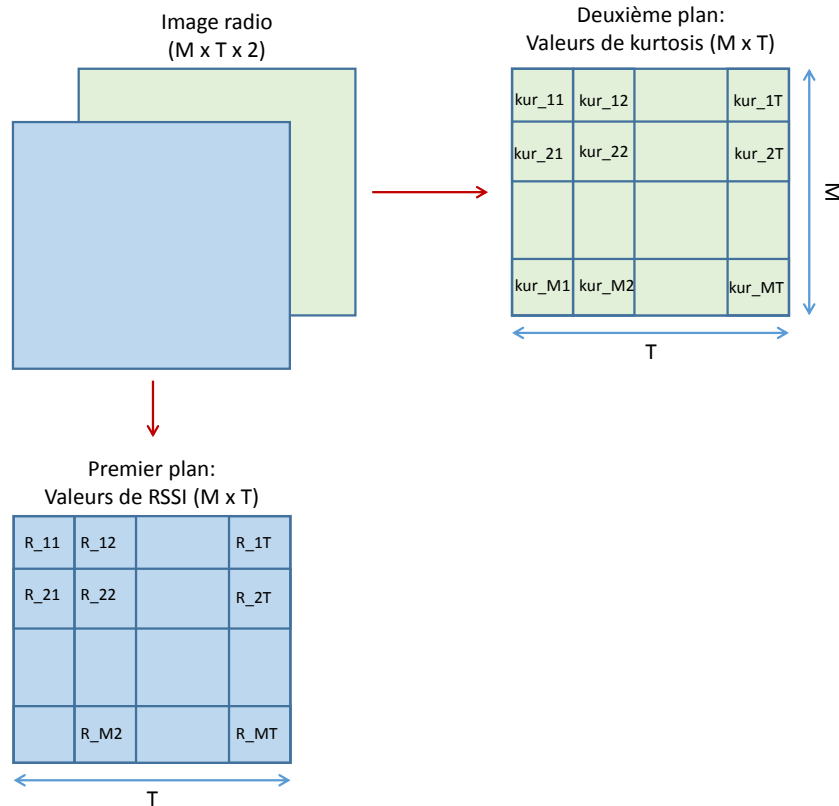


FIGURE 3.15: La structure des images radio.

3.5.3 Les modèles de réseaux de neurones convolutifs

Le réseau de neurones convolutif (CNN) fait partie des réseaux de neurones profonds (DNN), comprenant des couches différentes, chacune assurant une fonction spécifique. Un réseau de neurones convolutif conçu pour la reconnaissance de zone est constitué d'une ou de plusieurs couches de convolution suivies par une ou plusieurs couches entièrement connectées. Il prend des images radio en entrée et les étiquettes comme les neurones de sortie. Nous expliquons la fonction de chaque couche du CNN type présenté dans Figure 3.16.

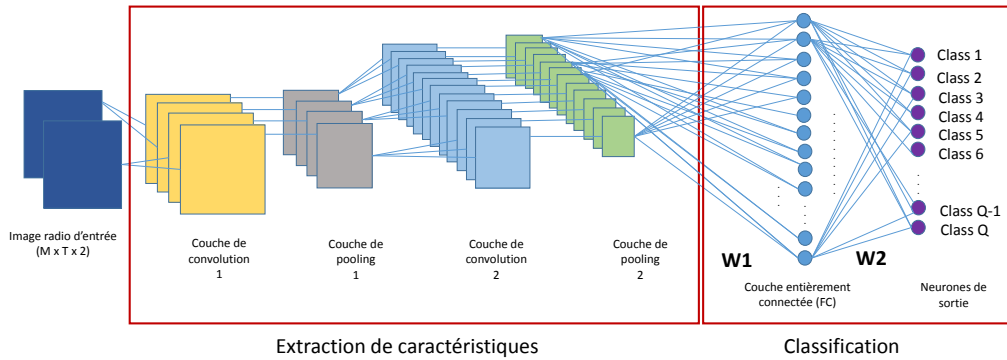


FIGURE 3.16: Un exemple d'architecture d'un réseau CNN ayant 2 couches de convolution et 1 couche entièrement connectée.

Comme indiqué précédemment, un CNN est constitué de plusieurs couches cachées situées entre les couches d'entrée et de sortie. Les couches cachées sont constituées de couches convolutives, de couches de pooling et de couches entièrement connectées. Le rôle de chaque couche est décrit en détail.

- **Couche de convolution** Après la couche d'entrée, qui prend les tenseurs radio, une structure typique de CNN a été conçue en commençant par un processus d'extraction de caractéristiques. Cette extraction des caractéristiques du tenseur d'entrée, est assurée par des filtres initialisés aléatoirement. Plusieurs filtres peuvent être utilisés pour extraire le maximum de caractéristiques contenues dans les données d'entrée. Après avoir fait glisser (convoluer) les filtres sur les pixels d'entrée, chaque sortie de convolution est transmise à une fonction d'activation. Le choix par défaut pour la fonction d'activation dans le CNN, la fonction 'unité linéaire rectifiée' (ReLU), a été utilisé. Cette fonction gère la non-linéarité des données. Elle est donnée par :

$$f_{ReLU}(x) = \begin{cases} x & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}, \quad (3.35)$$

où x est l'argument de la fonction (dans notre cas, x désigne chaque pixel de l'image radio en sortie d'une couche de convolution).

- **Couche de pooling** Cette couche est une couche de réduction spatiale qui sous-échantillonne les sorties de la couche de convolution précédente. Elle réduit la charge

de calcul et la complexité temporelle en réduisant la dimension des tenseurs obtenus en sortie de la couche convolutive précédente. Nous avons choisi la fonction max-pooling, qui sélectionne la valeur maximale de celles couvertes dans la fenêtre de regroupement actuelle (Figure 3.17). Lorsque nous utilisons un tenseur de petite taille et que nous voulons apprendre toutes les caractéristiques, cette couche peut être éliminée.

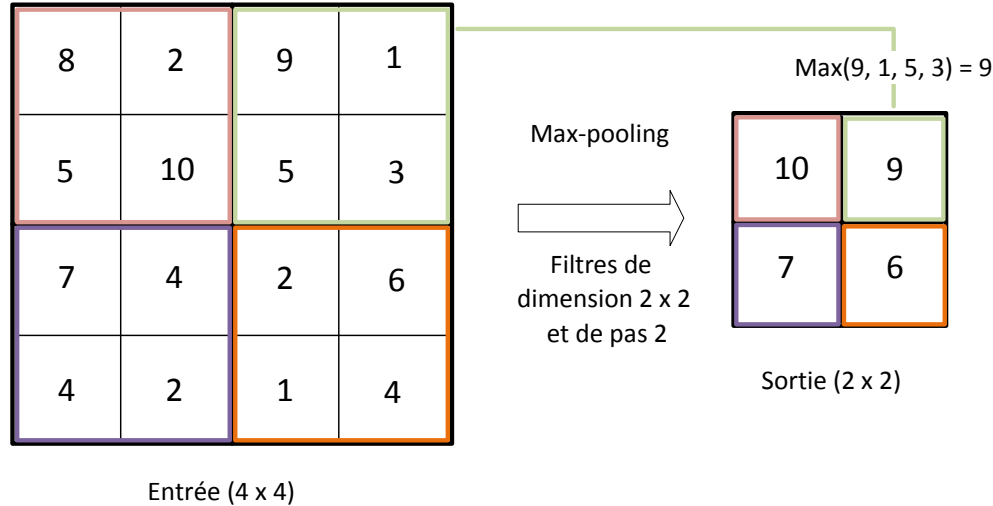


FIGURE 3.17: L'opération de max-pooling avec des images radio (fenêtre de taille (2×2) et un pas de 2).

- **Couche entièrement connectée (FC)** Après un processus d'extraction de caractéristiques fournis par une combinaison de couches convolutives et de couches de pooling, une couche entièrement connectée se charge d'identifier la probabilité associée à chaque classe à l'aide de la fonction softmax. La classe ayant la probabilité la plus élevée est considérée comme le résultat de sortie (dans notre cas, nos classes sont les partitions formées). Dans cette couche, les neurones sont tous connectés aux neurones de la couche précédente.

Le nombre et la taille des filtres, le nombre de couches (couches de convolution et couches entièrement connectées) et divers hyperparamètres de CNN sont déterminés par des simulations comme discutés ci-après. En effet, plusieurs architectures neuronales ont été testées afin de déterminer celle, la plus adéquate au problème donné. L'optimisation de différents paramètres est une étape cruciale du processus d'apprentissage de CNN, qui est un processus expérimental nécessitant plusieurs simulations.

Au cours de la phase d'apprentissage, notre modèle CNN utilise un algorithme de rétropropagation. Les poids sont itérativement mis à jour afin de minimiser la fonction de coût, entre la prédiction courante (estimation classe) et le libellé (classe réelle). L'algorithme qui a été choisi dans le cadre de cette thèse est ADAM car il a permis d'obtenir les meilleurs résultats dans le chapitre précédent. Afin d'éviter le phénomène de sur-apprentissage, nous avons introduit un taux de dropout. Ceci consiste à ignorer un sous-ensemble de neurones dans une couche donnée à l'apprentissage, c'est-à-dire en supprimant les noeuds de la couche à chaque étape d'apprentissage. Dans les modèles que nous proposons, le dropout est utilisé après chaque couche entièrement connectée.

Le taux de régularisation du dropout est mentionné dans chaque situation dans la suite du document.

3.5.4 Résultats : Evaluation des performances du Deep CNN

Dans cette section, nous présentons les différents résultats obtenus en justifiant expérimentalement le choix des différents paramètres utilisés, de l'algorithme d'optimisation et des architectures.

Environnement de simulation

Nous considérons un réseau de capteurs sans fil de M PAs (dans nos simulations, deux valeurs de M ont été utilisées, $M = 5$ et $M = 10$) et L points d'apprentissage, placés dans une zone de 400 m^2 (c'est-à-dire $20 \text{ m} \times 20 \text{ m}$). Cette zone est divisée en grilles de dimension $5 \text{ m} \times 5 \text{ m}$ ou $2 \text{ m} \times 2 \text{ m}$. Lorsqu'on travaille avec une grille de dimension $5 \text{ m} \times 5 \text{ m}$, on considère 4 points d'apprentissage par classe. Mais, on n'aura qu'un seul point d'apprentissage par classe, lorsque la dimension de la grille est de $2 \text{ m} \times 2 \text{ m}$.

Les emplacements des noeuds d'apprentissage et des PAs sont choisis aléatoirement dans la zone étudiée. La précision a été étudiée sur plusieurs réalisations d'environnement. Afin de simplifier la présentation des résultats, nous décrivons un environnement de test, comme illustré ci-après, où le choix de chaque paramètre est justifié expérimentalement. Différentes mesures RSSI ont été prélevées à chaque position d'apprentissage tout en tenant en compte du shadowing, en utilisant le modèle de propagation réel, mentionné dans le chapitre précédent et en effectuant plusieurs simulations, visant à simuler des conditions de propagation réelles.

La dimension de chaque fingerprint RSSI est $(M \times T)$ comme mentionné précédemment, où T varie pour choisir la meilleure valeur, comme nous le verrons plus tard. Le nombre de réalisations par point d'apprentissage est $N = 72$. Par conséquent, nous avons obtenu 72 tenseurs correspondant à chaque point d'apprentissage et la dimension de chaque tenseur est $(M \times T \times 1)$ avec la méthode CNNLocWoK et $(M \times T \times 2)$ lors de l'application de la méthode CNNLocWK. 80% des tenseurs ont été considérés durant la phase d'apprentissage et 20% pour la validation. Les performances présentées sont essentiellement basées sur les données de validation.

Les simulations ont été menées sur un PC doté d'un processeur Intel (R) Core (TM) i7-6700 à 3.4 GHz. MATLAB R2018a dispose d'un toolbox de réseau de neurones avancé, très efficace pour l'implémentation de nos modèles CNN.

Paramétrage et choix des hyperparamètres

L'optimisation des hyperparamètres et le choix de l'architecture CNN sont les facteurs les plus importants dans le développement de CNNLoc. L'identification des valeurs optimales des paramètres est définie par un processus expérimental. Ainsi, il a fallu plusieurs simulations. Les valeurs estimées dépendent des données d'entrée. Plusieurs architectures ont été déployées et testées (en variant le nombre de couches convolutives et les couches de pooling) afin de déterminer l'architecture optimale.

Comme l'apprentissage d'un modèle CNN est le processus le plus gourmand en terme de charge de calcul, nous avons utilisé des mini-lots de données (mini batches) d'entrée, plutôt que d'exposer la totalité des données, au réseau lors de l'apprentissage et ceci afin d'accélérer la convergence [130,131]. Par conséquent, une itération correspond au passage d'un mini lot de données. De cette manière, nous pouvons assurer une convergence plus robuste comparée à l'apprentissage par lots complets [132]. Une 'époque' est un passage complet de données. Le nombre maximal d'itérations pour atteindre la convergence est donné par l'équation 3.36. Le choix de la taille du mini-lot doit être bien étudié afin de diminuer le temps de convergence et assurer des meilleures performances. Par conséquent, la détermination de la taille appropriée du mini-lot est une étape cruciale dans les réseaux CNN.

$$\textit{itérations} = \frac{\textit{Données d'entrée}}{\textit{taille du mini batch}} \times \textit{époques} \quad (3.36)$$

Durant la phase d'apprentissage, plusieurs paramètres ont été ajustés (le nombre de filtres, la taille du filtre, la fenêtre et le pas du pooling, la taille du mini lot et le nombre des époques) afin d'obtenir l'architecture neuronale optimale. Pour les couches entièrement connectées, nous avons ajusté le nombre de neurones et le taux de dropout. Les paramètres illustrés dans tableau 3.9 sont les mêmes pour tous les modèles que nous utiliserons après. Pour les autres hyperparamètres non mentionnés (le nombre de filtres, la taille du mini-lot, le nombre d'époques, le nombre de neurones de la couche entièrement connectée, et le dropout), leurs valeurs dépendaient du modèle.

TABLE 3.9: Liste des hyperparamètres considérés.

Paramètre	Valeur
Nombre des neurones de sortie	16 (grille de dimension 5 m × 5 m) et 100 (grilles de dimension 2 m × 2 m)
Nombre des couches convolutives	0, 2, 3, 4, 5
Nombre des couches FC	1
Taille du filtre	2 × 2
Max pooling	Utilisé une seule fois après la première couche convolutive
Fenêtre du max pooling	2 avec un pas 1 ou 2
Paramètre T	2, 10, 20, 25, 30
Algorithme d'optimisation	Adam
Fonction d'activation	ReLU pour les couches de convolution et Softmax pour les couches FC

Evaluation de la précision de localisation

Dans la littérature, les approches de localisation basées sur l'apprentissage automatique peuvent résoudre des problèmes de régression ou des problèmes de classification. Dans ce

3.5 Localisation avec un modèle à base des réseaux de neurones convolutifs CNN 93

travail, motivé par la performance exceptionnelle de CNN dans la classification des images, nous avons traité la localisation en intérieur comme une reconnaissance de la zone où se trouve le noeud à localiser. Ainsi, en tant que sortie de notre modèle CNN, nous avons l'étiquette de la classe prédite contenant le noeud capteur à localiser, et non ses coordonnées de position. Pour évaluer les performances des algorithmes d'apprentissage, des métriques standards peuvent être calculées [133] en comparant les classes réelles et celles assignées. Nous avons introduit la précision en tant que métrique d'évaluation de performance de classification. C'est le pourcentage de noeuds capteurs correctement classés qui fait référence au taux de reconnaissance du classificateur. Cette métrique est définie comme :

$$précision = \frac{C_{true}}{C_{total}} \times 100 \quad (3.37)$$

Où C_{true} est le nombre de noeuds capteurs correctement classés et C_{total} est le nombre total des noeuds classés.

1. **Variation du paramètre T** : Comme indiqué précédemment, nous avons travaillé avec des bases de données de RSSI de dimension $(M \times T)$. Dans cette partie, nous étudions l'impact de la variation de T sur la précision de la localisation en tenant compte de la complexité et du temps d'apprentissage. Pour déterminer la meilleure précision de localisation et réduire le temps d'apprentissage, nous avons considéré 4 scénarios de simulation : 10 PAs et une grille de dimension 5 m \times 5 m, 5 PAs et une grille de dimension 2 m \times 2 m, 5 PAs et une grille de dimension 5 m \times 5 m et 10 PAs et une grille de dimension 2 m \times 2 m. Nous avons utilisé une architecture à deux couches de convolution et une couche totalement connectée (FC). La figure 3.18 illustre la précision de la localisation liée au paramètre T .

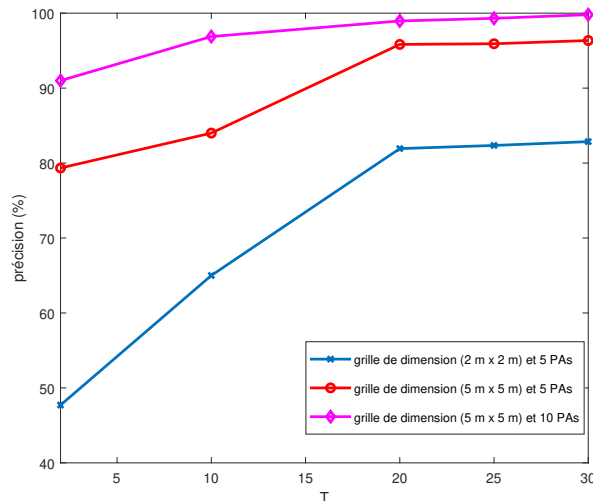


FIGURE 3.18: Variation de la précision en fonction du paramètre T .

Nous remarquons que la précision est devenue presque stable à partir de $T = 20$. Nous remarquons que le temps d'apprentissage a considérablement augmenté à partir de $T = 20$ (Tableaux 3.10 et 3.10). Le compromis entre précision et complexité nous

a amené à utiliser un paramètre T égal à 20. Nous mentionnons que les ressources en mémoire du PC de travail ne permettent pas de construire des bases de données RSSI avec $T > 20$ pour 10 PAs et des grilles de dimension 2 m \times 2 m. Cependant, étant donné que $T = 20$ s'est avéré être la meilleure valeur du paramètre T , comme le montre la figure 3.18, nous avons généralisé ce résultat approuvé pour les trois autres scénarios de simulation.

TABLE 3.10: Variation de la précision en fonction de T en utilisant une grille de dimension 5 m \times 5 m.

Nombre de PAs	Taille du mini batch	T	Précision (%)	Temps d'apprentissage (min)	Temps de prédiction (s)
5	60	2	79.34	0 :06	0.24×10^{-3}
		10	84	0 :07	0.29×10^{-3}
		20	95.83	0 :10	0.36×10^{-3}
		25	95.92	0 :50	0.83×10^{-3}
		30	96.35	1 :10	0.91×10^{-3}
10	60	2	91.57	0 :09	0.3×10^{-3}
		10	96.88	0 :09	0.48×10^{-3}
		20	98.96	0 :15	0.58×10^{-3}
		25	99.31	1 :23	1.3×10^{-3}
		30	99.88	1 :50	1.5×10^{-3}

TABLE 3.11: Variation de la précision en fonction de T en utilisant une grille de dimension 2 m \times 2 m.

Nombre de PAs	Taille du mini batch	T	Précision (%)	Temps d'apprentissage (min)	Temps de prédiction (s)
5	200	2	47.71	0 :13	0.17×10^{-3}
		10	65	0 :24	0.27×10^{-3}
		20	81.93	1 :25	0.3×10^{-3}
		25	82.35	4 :31	0.69×10^{-3}
		30	82.86	5 :15	0.93×10^{-3}
10	300	2	72.07	0 :47	0.2×10^{-3}
		10	85.36	1 :15	0.28×10^{-3}
		20	91.57	2 :52	0.68×10^{-3}
		25	X	—	—
		30	X	—	—

2. **Preuve empirique de l'utilisation bénéfique du kurtosis** : Nous démontrons l'avantage de l'utilisation du kurtosis sur la précision de localisation, de manière ex-

3.5 Localisation avec un modèle à base des réseaux de neurones convolutifs CNN 95

TABLE 3.12: La précision d'ADAM avec les données de validation en utilisant des grilles de dimension $2\text{ m} \times 2\text{ m}$ et 10 PAs.

	Taille du mini batch	Précision (%)	Nombre d'itérations
CNNLocWoK	300	91.57	285
CNNLocWK	400	94.13	190

périmentale. Nous voulions améliorer les performances d'Adam avec les fingerprints en introduisant un tenseur radio. Le tableau 3.12 illustre les performances obtenues lors de l'utilisation du kurtosis (CNNLocWK) et sans kurtosis (CNNLocWoK), sur les données de validation. Nous pouvons facilement remarquer que l'utilisation de ce paramètre garantit une amélioration de la précision de presque 2,5% et une accélération de la convergence de l'algorithme d'optimisation, qui a convergé au bout de 190 itérations au lieu de 285 itérations. Par conséquent, nous avons constaté que l'utilisation du kurtosis était un bon moyen pour améliorer la précision de localisation sans qu'il soit nécessaire de déployer une infrastructure supplémentaire. Les tests ont été effectués avec plusieurs scénarios, mais un seul est présenté dans ce rapport.

- 3. Architecture CNN : Variation du nombre des couches de convolution** Pour analyser l'impact du nombre de couches, différents modèles de CNN ont été réalisés avec des nombres différents des couches convolutives et des nombres différents de neurones dans chaque couche. La taille du mini batch considérée est de 300. Nous avons utilisé une couche entièrement connectée avec 120 neurones et un taux de dropout de l'ordre de 0.3 après le module d'extraction de caractéristiques. La couche de sortie est composée des neurones de sortie (100 classes parce que nous avons travaillé avec une grille de dimension $2\text{ m} \times 2\text{ m}$). Les résultats sont résumés dans le tableau 3.13. Différents modèles ont été testés afin de trouver le meilleur nombre de filtres dans chaque couche. Pour les filtres, nous avons utilisé le zéro padding avec un pas égal à 1. On note que : $\text{Conv}(p, q)$ la couche de convolution avec p filtres de dimension $(q \times q)$ et $\text{Maxpooling}(y, z)$ est la couche de max-pooling de fenêtre $(y \times y)$ et de pas z .

Il est clair que le réseau CNN à deux couches convolutives, est associé à la meilleure précision et offre des performances meilleures que les autres. Ceci est basé sur le fait qu'un tel réseau est assez complexe pour extraire les caractéristiques appropriées pour la reconnaissance de zone. Un réseau CNN avec plus que deux couches convolutives est un modèle complexe tendant à provoquer un sur-apprentissage. Par conséquent, le déploiement d'un réseau CNN à deux couches semble être la meilleure architecture pour obtenir une bonne précision.

- 4. Comparaison de la précision de localisation en intérieur de différentes approches** La méthode de localisation en intérieur proposée est basée sur un CNN en utilisant les valeurs RSSI. Elle peut être évaluée et comparée aux méthodes standards. Toutes les méthodes comparées ici utilisent les informations RSSI pour localiser un noeud de capteur. La technique de la trilatération est basée sur les distances entre le noeud à localiser et les PAs, nécessitant au moins trois distances connues. Basé sur le NN traditionnel, nous avons introduit deux systèmes "Classic NN" et "Classic NN2".

TABLE 3.13: Variation de la précision en fonction du nombre de couches en utilisant une grille de dimension $2\text{ m} \times 2\text{ m}$ et 10 PAs.

Nombre de couches de convolution	Précision (%)	Architecture du module d'extraction de caractéristiques
0	83.3	–
2	91.57	Conv(200,2) Maxpooling(2,2) Conv(120,2)
3	88.43	Conv(120,2) Maxpooling(2,2) Conv(200,2) Conv(300,2)
4	83.29	Conv(40,2) Maxpooling(2,2) Conv(90,2) Conv(300,2) Conv(400,2)
5	82	Conv(40,2) Maxpooling(2,2) Conv(90,2) Conv(300,2) Conv(400,2) Conv(700,2)

Classic NN est un réseau composé de 5 couches FC ayant le même ordre de complexité, en termes de nombre de poids à apprendre, enregistré par CNN, afin de comparer ces techniques équitablement. Classic NN2 est vérifié expérimentalement comme étant le meilleur modèle NN mis en oeuvre sur la base de nos données. Pour y arriver, nous avons commencé avec un modèle d'une seule FC. La meilleure précision de localisation de 83.3% est obtenue avec 120 neurones. Ensuite, un modèle à deux couches FC est construit. Le nombre de neurones est optimisé pour atteindre 84.5% de précision. Une troisième couche FC est ajoutée, contribuant à une précision de 84.75%. La précision a diminué pour atteindre 80% avec 4 couches FC. Par conséquent, nous avons travaillé avec le modèle à trois couches FC, qui avait la meilleure précision. Le tableau 3.15 présente la précision de localisation associée à chaque technique, afin de comparer les performances de nos frameworks de localisation développés (CNNLocWoK et CNN-LocWK) et d'autres approches existantes qui n'utilisent pas l'information de Kurtosis (trilatération, Classic NN, et Classic NN2). Les architectures des modèles associés aux résultats fournis sont présentés dans le tableau 3.14.

Nous remarquons facilement que la trilatération donne une mauvaise "précision" par rapport aux autres algorithmes testés. Le NN classique donne une bonne précision,

3.5 Localisation avec un modèle à base des réseaux de neurones convolutifs CNN 97

TABLE 3.14: Les architectures des modèles de deep learning utilisés.

Algorithme de deep learning	Architecture du réseau
Classic NN	FC(1500) FC(3000) FC(2000) FC(1200) FC(120)
Classic NN2	FC(100) FC(200) FC(120)
CNNLocWoK	Conv(200,2) Maxpooling(2,2) Conv(120,2) FC(120)
CNNLocWK	Conv(200,2) Maxpooling(2,2) Conv(300,2) FC(120)

mais il est moins précis que les systèmes de localisation en intérieur basés sur CNN. Pour atteindre une bonne précision, nous avons dû modéliser correctement le réseau car nous pourrions obtenir une meilleure précision avec une complexité moindre (Classic NN2 est meilleur et moins complexe que Classic NN), d'autant plus que les tenseurs radio considérés ne sont pas volumineux. CNN offre une précision importante de localisation, lorsqu'il est utilisé avec kurtosis, seulement 5.84% des classes ont été mal estimées, dont 96,75% ont été estimées comme classe juste voisine. Nous notons que cette erreur de classification a toujours été causée par des points d'apprentissage situés près des frontières séparant les classes. Nous constatons que la trilatération combinée avec la complétion de matrice (Trilatération-MC) offre la meilleure précision de localisation. Cependant, en termes de complexité de calcul, c'est CNNLocWK qui l'emporte

TABLE 3.15: Comparaison de la précision associée aux différents algorithmes en utilisant des grilles de dimension 2 m × 2 m et 10 PAs.

Technique de localisation en intérieur	Précision (%)
Trilateration	30
Trilatération-MC	95
Classic NN	80.76
Classic NN2	84.75
CNNLocWoK	91.57
CNNLocWK	94.13

puisqu'il assure une localisation en 10^{-3} s et décale toute la complexité online en offline, contrairement à Trilatération-MC où tout le processus est online.

3.6 Conclusion

Exploitant les RSSI fingerprints de la base réelle UJIndoorLoc, nous avons réussi à mettre en place une étude comparative de différentes méthodes afin de trouver la meilleure combinaison entre évaluation de similarité et estimation de position. Nous avons obtenu des résultats satisfaisants en terme de précision de localisation. Cependant, le fait d'appliquer des formules mathématiques parfois complexes en ligne peut engendrer une complexité de calcul élevée et augmenter le temps de prédiction de la position. Or, pour évaluer les performances d'un système de localisation, il y a un compromis entre précision et complexité.

Pour faire face à un tel problème, la construction d'un système optimisé, en phase offline, pour prédire la position d'un utilisateur en online semble être une bonne solution. Pour cela, nous avons développé une relation analytique entre les puissances reçues en un point et ses coordonnées en obtenant un système linéaire qui permet de localiser un utilisateur donné. Les résultats obtenus par simulations ont montré qu'en appliquant ce système, nous obtenons des bons résultats en terme de précision de localisation et nous décalons toute la complexité online en offline. Cependant, un tel système n'est pas toujours accessible vu les contraintes et les hypothèses que nous devons respecter afin de le construire. Nous avons donc eu recours à une solution basée sur l'apprentissage.

Pour cela, une solution de localisation en intérieur basée sur CNN et des mesures RSSI a été développée. Cette méthode exploite non seulement les mesures RSSI, mais également les valeurs de kurtosis correspondantes associées, visant à fournir de nouvelles informations au réseau. Nous avons converti des tenseurs radio contenant des valeurs RSSI, reçues pendant un temps T et provenant de différents PAs et les avons associés aux valeurs de kurtosis calculées et organisées sous la forme d'images radio de 3 dimensions afin de réaliser une reconnaissance de zone. Cette méthode résout le problème de la grande complexité de calcul des méthodes traditionnelles et garantit une bonne précision de localisation. Comme mentionné précédemment, l'optimisation des hyperparamètres et des différentes architectures utilisées est le facteur le plus important dans l'optimisation des performances du CNN. Pour identifier les valeurs optimales de différents paramètres, plusieurs expérimentations ont été effectuées. Ce processus expérimental a démontré qu'un modèle CNN à deux couches de convolutions, optimisées par Adam est le plus adapté pour ce problème de localisation. Les résultats de simulation ont prouvé que notre approche est robuste et meilleure que les autres méthodes classiques mentionnées, considérant le compromis entre précision de localisation et complexité de calcul.

Conclusions et perspectives

Conclusions

La motivation principale de cette thèse a été de proposer des améliorations aux techniques de localisation existantes et de les adapter aux différents scénarios de communications existants. Une première partie de la thèse a été consacrée à une étude approfondie des différentes notions intervenant dans la localisation ainsi que des principales techniques existantes.

Nous avons ainsi identifié deux axes de recherches sur lesquels la thèse a été construite. Le premier axe concerne l'information utilisée pour l'estimation de la position. Le second axe quant à lui concerne l'algorithme qui va être appliqué pour l'estimation de la position. Dans le cadre de cette thèse, le choix s'est porté sur l'utilisation des puissances reçues par l'objet à localiser. Dans le premier axe de recherche, en utilisant une base WiFi existante, la contribution de la thèse a été de proposer une stratégie de sélection des points d'accès dont la puissance reçue constitue une information pertinente pour la localisation. Le critère de sélection adopté a été l'entropie. Pour cette partie, un algorithme de localisation basé sur la technique des K plus proches voisins a été utilisé pour montrer qu'il a été possible d'arriver à une même précision de localisation en utilisant moins de points d'accès et ainsi réduire le temps de mesure et le temps de calcul pendant la phase en ligne. En ce qui concerne les algorithmes de localisation, nous nous sommes intéressés aux deux familles de techniques suivantes :

- La multi-latération qui consiste à estimer la position d'un objet en se basant sur la connaissance de la distance qui le sépare de plusieurs positions connues. Ces dernières sont calculées à partir d'un modèle de propagation et les mesures des puissances reçues. Les conditions de propagation dans un milieu intérieur peuvent rendre le niveau de puissances reçues inférieur à la sensibilité du récepteur et ainsi la distance ne peut pas être calculée. Nous avons proposé dans le cadre de cette thèse, un algorithme qui permet de compléter une matrice des distances entre les différents noeuds communicants d'un réseau en se basant sur la connaissance de certaines distances et des algorithmes d'optimisation. Ainsi, la précision de localisation a été améliorée.
- Le fingerprinting : L'étude a commencé par une comparaison des différentes métriques d'évaluation de la similarité. Une telle solution est coûteuse en termes de stockage et de calcul. Afin de transférer cette complexité à la phase hors ligne, on propose de construire une relation analytique entre les puissances reçues et les coordonnées de l'objet en utilisant le développement de Taylor. En raison de la difficulté de trouver une relation linéaire, un réseau de neurones convolutif a été utilisé en formulant la localisation en un problème de classification. Il a donc fallu mettre en forme les puissances

reçues afin de les présenter à l'entrée du réseau de neurones, choisir l'architecture et l'algorithme d'apprentissage adéquat. Les résultats obtenus ont montré des bonnes performances de localisation qui peuvent être obtenues.

Le travail de cette thèse a donc permis de proposer une solution de localisation adaptée selon l'architecture du réseau en minimisant la complexité ou en la transférant vers une phase hors ligne. Ceci rend les solutions proposées applicables dans un contexte d'Internet des objets.

Perspectives

Parmi les perspectives, nous pourrions proposer les actions suivantes :

- Construire notre propre base de mesures réelles en utilisant une configuration d'environnement intérieur sur laquelle nous pouvons appliquer tous nos systèmes développés.
- Dans nos travaux, nous n'avons pas étudié systématiquement l'impact du bruit sur les mesures. Dans les travaux futurs, nous pourrions mener une étude approfondie sur les différents types de bruit et intégrer un processus de minimisation de son effet dans les systèmes proposés.
- Approfondir la localisation à base des CNNs. Pour cela, nous pourrions utiliser des informations supplémentaires telles que celles provenant des capteurs inertiels, dans le but d'améliorer les résultats obtenus en exploitant uniquement les valeurs de RSSI.

Bibliographie

- [1] Rathin Chandra Shit, Suraj Sharma, Deepak Puthal, and Albert Y Zomaya, “Location of things (lot) : A review and taxonomy of sensors localization in iot infrastructure,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2028–2061, 2018.
- [2] Luigi Atzori, Antonio Iera, and Giacomo Morabito, “The internet of things : A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] S Vuppala et al., “ubiquitous, secure internet-of-things with location and context-awareness,” *BUTLER project D*, vol. 2, pp. 1–171.
- [4] Domenico Ciuonzo, Pierluigi Salvo Rossi, and Peter Willett, “Generalized rao test for decentralized detection of an uncooperative target,” *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 678–682, 2017.
- [5] Bram Dil, Stefan Dulman, and Paul Havinga, “Range-based localization in mobile sensor networks,” in *European Workshop on Wireless Sensor Networks*. Springer, 2006, pp. 164–179.
- [6] Santar Pal Singh and SC Sharma, “Range free localization techniques in wireless sensor networks : A review,” *Procedia Computer Science*, vol. 57, pp. 7–16, 2015.
- [7] S Hamed Javadi, Hossein Moosaei, and Domenico Ciuonzo, “Learning wireless sensor networks for source localization,” *Sensors*, vol. 19, no. 3, pp. 635, 2019.
- [8] Nirupama Bulusu, John Heidemann, Deborah Estrin, et al., “Gps-less low-cost outdoor localization for very small devices,” *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [9] Jun-Ho Huh and Kyungryoung Seo, “An indoor location-based control system using bluetooth beacons for iot systems,” *Sensors*, vol. 17, no. 12, pp. 2917, 2017.
- [10] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A Al-Ammar, and Hend S Al-Khalifa, “Ultra wideband indoor positioning technologies : Analysis and recent advances,” *Sensors*, vol. 16, no. 5, pp. 707, 2016.
- [11] Zengqiang Chen and Chen Wang, “Modeling rfid signal distribution based on neural network combined with continuous ant colony optimization,” *Neurocomputing*, vol. 123, pp. 354–361, 2014.
- [12] Ahmed Shokry, Moustafa Elhamshary, and Moustafa Youssef, “The tale of two localization technologies : Enabling accurate low-overhead wifi-based localization for low-end phones,” in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2017, p. 42.

- [13] Xiaomeng Zhang, Hao Sun, Shuopeng Wang, and Jing Xu, "A new regional localization method for indoor sound source based on convolutional neural networks," *IEEE Access*, vol. 6, pp. 72073–72082, 2018.
- [14] Yingxiang Sun, Jiajia Chen, Chau Yuen, and Susanto Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6403–6413, 2017.
- [15] Mitesh Patel, Brendan Emery, and Yan-Ying Chen, "Contextualnet : Exploiting contextual information using lstms to improve image-based localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.
- [16] Ling Wu, Chi-Hua Chen, and Qishan Zhang, "A mobile positioning method based on deep learning techniques," *Electronics*, vol. 8, no. 1, pp. 59, 2019.
- [17] Rizanne Elbakly, Heba Aly, and Moustafa Youssef, "Truestory : Accurate and robust rf-based floor estimation for challenging indoor environments," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10115–10124, 2018.
- [18] Wenhua Shao, Haiyong Luo, Fang Zhao, Yan Ma, Zhongliang Zhao, and Antonino Crivello, "Indoor positioning based on fingerprint-image and deep learning," *IEEE Access*, vol. 6, pp. 74699–74712, 2018.
- [19] Tianyi Zeng, Yongyu Chang, Qi Zhang, Mengshi Hu, and Junyao Li, "Cnn-based los/nlos identification in 3-d massive mimo systems," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2491–2494, 2018.
- [20] Chenwei Cai, Li Deng, Mingyang Zheng, and Shufang Li, "Pilc : Passive indoor localization based on convolutional neural networks," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. IEEE, 2018, pp. 1–6.
- [21] Dragos Niculescu and Badri Nath, "Ad hoc positioning system (aps) using aoa," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. Ieee, 2003, vol. 3, pp. 1734–1743.
- [22] Tufan C Karalar and Jan Rabaey, "An rf tof based ranging implementation for sensor networks," in *2006 IEEE International Conference on Communications*. IEEE, 2006, vol. 7, pp. 3347–3352.
- [23] Xiuzhen Cheng, Andrew Thaeler, Guoliang Xue, and Dechang Chen, "Tps : A time-based positioning scheme for outdoor wireless sensor networks," in *IEEE INFOCOM 2004*. IEEE, 2004, vol. 4, pp. 2685–2696.
- [24] Zheng Yang and Yunhao Liu, "Quality of trilateration : Confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2009.
- [25] Abdo Y Alfakih, Amir Khandani, and Henry Wolkowicz, "Solving euclidean distance matrix completion problems via semidefinite programming," *Computational optimization and applications*, vol. 12, no. 1-3, pp. 13–30, 1999.
- [26] Yu Gu, Mengni Chen, Fuji Ren, and Jie Li, "Hed : Handling environmental dynamics in indoor wifi fingerprint localization," in *2016 IEEE wireless communications and networking conference*. IEEE, 2016, pp. 1–6.

- [27] Salah Eddine Elayoubi, Mikael Fallgren, Panagiotis Spapis, Gerd Zimmermann, David Martín-Sacristán, Changqing Yang, Sébastien Jeux, Patrick Agyapong, Luis Campoy, Yinan Qi, et al., “5g service requirements and operational use cases : Analysis and metis ii vision,” in *2016 European Conference on Networks and Communications (EuCNC)*. IEEE, 2016, pp. 158–162.
- [28] Jose Luis V Carrera, Zhongliang Zhao, Torsten Braun, Haiyong Luo, and Fang Zhao, “Discriminative learning-based smartphone indoor localization,” *arXiv preprint arXiv :1804.03961*, 2018.
- [29] Eric A Prigge et al., *A positioning system with no line-of-sight restrictions for cluttered environments*, Ph.D. thesis, Stanford University Stanford, CA, USA :, 2004.
- [30] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta, “Luxapose : Indoor positioning with mobile phones and visible light,” in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 447–458.
- [31] Antonio Costanzo and Valeria Loscri, “Système de localisation à lumière visible avec compensation du bruit environnemental,” 2019.
- [32] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, Xufei Mao, Panlong Yang, Yunhao Liu, and Xingfu Wang, “Swadloon : Direction finding and indoor localization using acoustic signal by shaking smartphones,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2145–2157, 2014.
- [33] Cristina V Lopes, Amir Haghighat, Atri Mandal, Tony Givargis, and Pierre Baldi, “Localization of off-the-shelf mobile devices using audible sound : architectures, protocols and performance assessment,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 2, pp. 38–50, 2006.
- [34] Atri Mandal, Cristina V Lopes, Tony Givargis, Amir Haghighat, Raja Jurdak, and Pierre Baldi, “Beep : 3d indoor positioning using audible sound,” in *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*. IEEE, 2005, pp. 348–353.
- [35] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik, “Indoor localization without infrastructure using the acoustic background spectrum,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 155–168.
- [36] Wael Elloumi, *Contributions à la localisation de personnes par vision monoculaire embarquée*, Ph.D. thesis, Université d’Orléans, 2012.
- [37] Mike Hazas and Andy Hopper, “Broadband ultrasonic location systems for improved indoor positioning,” *IEEE Transactions on mobile Computing*, vol. 5, no. 5, pp. 536–547, 2006.
- [38] Nissanka Bodhi Priyantha, *The cricket indoor location system*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.
- [39] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan, “The cricket location-support system,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 32–43.

- [40] Harsh Sinha, Jay Patrikar, Eeshan Gunesh Dhekane, Gaurav Pandey, and Mangal Kothari, "Convolutional neural network based sensors for mobile robot relocalization," in *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*. IEEE, 2018, pp. 774–779.
- [41] Ren C Luo and Wei Shih, "Autonomous mobile robot intrinsic navigation based on visual topological map," in *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2018, pp. 541–546.
- [42] Naoki Akail, Luis Yoichi Morales, and Hiroshi Murase, "Reliability estimation of vehicle localization result," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 740–747.
- [43] "<https://www.directindustry.fr/prod/enova-robotics/product-177187-1764197.html>," (accessed 30-06-2019).
- [44] Jó Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle, "Footpath : Accurate map-based indoor navigation using smartphones," in *2011 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2011, pp. 1–8.
- [45] Paula Tarrío, Matteo Cesana, Marco Tagliasacchi, Alessandro Redondi, Luca Borsani, and José R Casar, "An energy-efficient strategy for combined rss-pdr indoor localization," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2011, pp. 619–624.
- [46] Faheem Zafari, Ioannis Papapanagiotou, and Konstantinos Christidis, "Microlocation for internet-of-things-equipped smart buildings," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 96–112, 2015.
- [47] "Insiteo. <http://www.insiteo.com/joomla/index.php/fr/>," (accessed 15-06-2019).
- [48] "Polestar. <http://www.polestar.eu/fr/>," (accessed 15-06-2019).
- [49] Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu, "Wireless sensor networks : A survey on the state of the art and the 802.15.4 and zigbee standards," *Computer communications*, vol. 30, no. 7, pp. 1655–1695, 2007.
- [50] Sverre Holm, "Hybrid ultrasound-rfid indoor positioning : Combining the best of both worlds," in *2009 IEEE International Conference on RFID*. IEEE, 2009, pp. 155–162.
- [51] Mathieu Bouet and Aldri L Dos Santos, "Rfid tags : Positioning principles and localization techniques," in *2008 1st IFIP Wireless Days*. Ieee, 2008, pp. 1–5.
- [52] T Sanpechuda and L Kovavisaruch, "A review of rfid localization : Applications and techniques," in *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. IEEE, 2008, vol. 2, pp. 769–772.
- [53] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [54] Frank Viquez, Anna Lee Dragon, and Tim Archdeacon, "Location based services, a strategic analysis of wireless technologies, markets," Tech. Rep., and trends. Technical report, Allied Business Intelligence.

- [55] Allen Noguee, "Ready or not, mobile location technology is here," *Cahners In-Stat Group*, 2001.
- [56] Emma France, Karima Tracey, Geog Neureiter, Riccardo Pascotto, Miguel Serrano, Seppo Parkkila, and Juha Pirinen, "Where are the other mobile buddies around town," *IST Summit Mobile*, 2001.
- [57] Ana Roxin, Jaafar Gaber, Maxime Wack, and A Nait-Sidi-Moh, "Survey of wireless geolocation techniques," in *2007 IEEE Globecom Workshops*. IEEE, 2007, pp. 1–9.
- [58] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi, "Long-range communications in unlicensed bands : The rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.
- [59] Toni Adame, Albert Bel, Boris Bellalta, Jaume Barcelo, and Miquel Oliver, "Ieee 802.11 ah : the wifi approach for m2m communications," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 144–152, 2014.
- [60] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, et al., "Place lab : Device positioning using radio beacons in the wild," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 116–133.
- [61] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 233–245.
- [62] Dragos Niculescu and Badri Nath, "Ad hoc positioning system (aps) using aoa," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. Ieee, 2003, vol. 3, pp. 1734–1743.
- [63] Ismail Guvenc and Chia-Chin Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.
- [64] Yang Tian, Ruipeng Gao, Kaigui Bian, Fan Ye, Tao Wang, Yizhou Wang, and Xiaoming Li, "Towards ubiquitous indoor localization service leveraging environmental physical features," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 55–63.
- [65] Tufan C Karalar and Jan Rabaey, "An rf tof based ranging implementation for sensor networks," in *2006 IEEE International Conference on Communications*. IEEE, 2006, vol. 7, pp. 3347–3352.
- [66] Xiuzhen Cheng, Andrew Thaeler, Guoliang Xue, and Dechang Chen, "Tps : A time-based positioning scheme for outdoor wireless sensor networks," in *IEEE INFOCOM 2004*. IEEE, 2004, vol. 4, pp. 2685–2696.
- [67] Praveen Kumar, Lohith Reddy, and Shirshu Varma, "Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks," in *2009 Fifth international conference on wireless communication and sensor networks (WCSN)*. IEEE, 2009, pp. 1–4.

- [68] Linqing Gui, *Improvement of range-free localization systems in wireless sensor networks*, Ph.D. thesis, Toulouse, INSA, 2013.
- [69] Rejane Dalce, *Méthodes de localisation par le signal de communication dans les réseaux de capteurs sans fil en intérieur*, Ph.D. thesis, Toulouse, INSA, 2013.
- [70] Massimo Ficco, Francesco Palmieri, and Aniello Castiglione, “Hybrid indoor and outdoor location services for new generation mobile terminals,” *Personal and ubiquitous computing*, vol. 18, no. 2, pp. 271–285, 2014.
- [71] Zheng Yang and Yunhao Liu, “Quality of trilateration : Confidence-based iterative localization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2009.
- [72] Abdelmoumen Norrdine, “An algebraic solution to the multilateration problem,” in *Proceedings of the 15th international conference on indoor positioning and indoor navigation, Sydney, Australia*, 2012, vol. 1315.
- [73] Thu Nguyen and Yoan Shin, “Matrix completion optimization for localization in wireless sensor networks for intelligent iot,” *Sensors*, vol. 16, no. 5, pp. 722, 2016.
- [74] Ricardo Ferreira, João Xavier, João P Costeira, and Victor Barroso, “Newton algorithms for riemannian distance related problems on connected locally symmetric manifolds,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 4, pp. 634–645, 2013.
- [75] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [76] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma, “Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix,” *Coordinated Science Laboratory Report no. UILU-ENG-09-2214, DC-246*, 2009.
- [77] Zhouchen Lin, Minming Chen, and Yi Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *arXiv preprint arXiv :1009.5055*, 2010.
- [78] Chunsheng Liu, Hong Shan, and Bin Wang, “Wireless sensor network localization via matrix completion based on bregman divergence,” *Sensors*, vol. 18, no. 9, pp. 2974, 2018.
- [79] Luong Nguyen and Byonghyo Shim, “Localization of internet of things network via euclidean distance matrix completion,” in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2016, pp. 1–4.
- [80] Fu Xiao, Wei Liu, Zhetao Li, Lei Chen, and Ruchuan Wang, “Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2017.
- [81] Thu Nguyen and Yoan Shin, “Matrix completion optimization for localization in wireless sensor networks for intelligent iot,” *Sensors*, vol. 16, no. 5, pp. 722, 2016.
- [82] Michael Grant, Stephen Boyd, and Yinyu Ye, “Cvx : Matlab software for disciplined convex programming,” 2008.

- [83] Xiansheng Guo, Lei Chu, and Xiang Sun, "Accurate localization of multiple sources using semidefinite programming based on incomplete range matrix," *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5319–5324, 2016.
- [84] Xiansheng Guo, Lei Chu, and Nirwan Ansari, "Joint localization of multiple sources from incomplete noisy euclidean distance matrix in wireless networks," *Computer Communications*, vol. 122, pp. 20–29, 2018.
- [85] Sinem Bozkurt, Guln Elibol, Serkan Gunal, and Ugur Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, 2015, pp. 1–8.
- [86] Xiansheng Guo, Lei Chu, and Nirwan Ansari, "Joint localization of multiple sources from incomplete noisy euclidean distance matrix in wireless networks," *Computer Communications*, vol. 122, pp. 20–29, 2018.
- [87] Junting Chen and Urbashi Mitra, "Unimodality-constrained matrix factorization for non-parametric source localization," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2371–2386, 2019.
- [88] Luong T Nguyen, Junhan Kim, Sangtae Kim, and Byonghyo Shim, "Localization of iot networks via low-rank matrix completion," *IEEE Transactions on Communications*, 2019.
- [89] Warren S Torgerson, "Multidimensional scaling of similarity," *Psychometrika*, vol. 30, no. 4, pp. 379–393, 1965.
- [90] Margrit Betke and Leonid Gurvits, "Mobile robot localization using landmarks," *IEEE transactions on robotics and automation*, vol. 13, no. 2, pp. 251–263, 1997.
- [91] Faheem Zafari, Ioannis Papanagiotou, and Konstantinos Christidis, "Microlocation for internet-of-things-equipped smart buildings," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 96–112, 2015.
- [92] Moustafa Youssef and Ashok Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.
- [93] Paramvir Bahl, Venkata N Padmanabhan, Victor Bahl, and Venkat Padmanabhan, "Radar : An in-building rf-based user location and tracking system," 2000.
- [94] Beenish A Akram, Ali H Akbar, and Omair Shafiq, "Hybloc : Hybrid indoor wi-fi localization using soft clustering-based random decision forest ensembles," *IEEE Access*, vol. 6, pp. 38251–38272, 2018.
- [95] Rizzanne Elbakly, Heba Aly, and Moustafa Youssef, "Truestory : Accurate and robust rf-based floor estimation for challenging indoor environments," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10115–10124, 2018.
- [96] Xiaomeng Zhang, Hao Sun, Shuopeng Wang, and Jing Xu, "A new regional localization method for indoor sound source based on convolutional neural networks," *IEEE Access*, vol. 6, pp. 72073–72082, 2018.
- [97] Yingxiang Sun, Jiajia Chen, Chau Yuen, and Susanto Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6403–6413, 2017.

- [98] Ricardo Leonardo, Marilia Barandas, and Hugo Gamboa, "A framework for infrastructure-free indoor localization based on pervasive sound analysis," *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4136–4144, 2018.
- [99] Shin-Jye Lee, Tonglin Chen, Lun Yu, and Chin-Hui Lai, "Image classification based on the boost convolutional neural network," *IEEE Access*, vol. 6, pp. 12755–12768, 2018.
- [100] Mai Ibrahim, Marwan Torki, and Mustafa ElNainay, "Cnn based indoor localization using rss time-series," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 01044–01049.
- [101] Baoding Zhou, Jun Yang, and Qingquan Li, "Smartphone-based activity recognition for indoor localization using a convolutional neural network," *Sensors*, vol. 19, no. 3, pp. 621, 2019.
- [102] Xiaomeng Zhang, Hao Sun, Shuopeng Wang, and Jing Xu, "A new regional localization method for indoor sound source based on convolutional neural networks," *IEEE Access*, vol. 6, pp. 72073–72082, 2018.
- [103] Jin-Woo Jang and Song-Nam Hong, "Indoor localization with wifi fingerprinting using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 753–758.
- [104] Abhinav Valada, Noha Radwan, and Wolfram Burgard, "Deep auxiliary learning for visual localization and odometry," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6939–6946.
- [105] Ren C Luo and Wei Shih, "Autonomous mobile robot intrinsic navigation based on visual topological map," in *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2018, pp. 541–546.
- [106] Hao Chen, Yifan Zhang, Wei Li, Xiaofeng Tao, and Ping Zhang, "Confi : Convolutional neural networks based indoor wi-fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.
- [107] Michael Tsai, "Path-loss and shadowing (large-scale fading)," *National Taiwan University October*, vol. 20, pp. 2011, 2011.
- [108] Emmanuel J Candès and Benjamin Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717, 2009.
- [109] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv :1609.04747*, 2016.
- [110] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.
- [111] Yiwen Wang, Sung-Phil Kim, and Jose C Principe, "Comparison of tdnn training algorithms in brain machine interfaces," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*. IEEE, 2005, vol. 4, pp. 2459–2462.
- [112] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

- [113] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” *Cited on*, vol. 14, pp. 8, 2012.
- [114] Diederik P Kingma and Jimmy Ba, “Adam : A method for stochastic optimization,” *arXiv preprint arXiv :1412.6980*, 2014.
- [115] Volker Strassen, “Gaussian elimination is not optimal,” *Numerische mathematik*, vol. 13, no. 4, pp. 354–356, 1969.
- [116] Anne Benoit, “Algorithmique i-cours et travaux dirigés l3, ecole normale supérieure de lyon,” 2009.
- [117] Bruno Iannazzo, “On the newton method for the matrix p th root,” *SIAM journal on matrix analysis and applications*, vol. 28, no. 2, pp. 503–523, 2006.
- [118] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché, “A comparative survey of wlan location fingerprinting methods,” in *2009 6th workshop on positioning, navigation and communication*. IEEE, 2009, pp. 243–251.
- [119] Adriano Moreira, Maria João Nicolau, Filipe Meneses, and António Costa, “Wi-fi fingerprinting in the real world-rtls@ um at the evaal competition,” in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2015, pp. 1–10.
- [120] Sinem Bozkurt, Gulim Elibol, Serkan Gunal, and Ugur Yayan, “A comparative study on machine learning algorithms for indoor positioning,” in *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2015, pp. 1–8.
- [121] D Zimmermann, J Baumann, A Layh, F Landstorfer, R Hoppe, and G Wolffe, “Database correlation for positioning of mobile terminals in cellular networks using wave propagation models,” in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*. IEEE, 2004, vol. 7, pp. 4682–4686.
- [122] Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez-Usó, Joan P Avariento, Tomás J Arnau, Mauri Benedito-Bordonau, and Joaquín Huerta, “Ujiindoorloc : A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems,” in *2014 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 2014, pp. 261–270.
- [123] Shih-Hau Fang and Tsungnan Lin, “Principal component localization in indoor wlan environments,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, 2011.
- [124] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché, “A comparative survey of wlan location fingerprinting methods,” in *2009 6th workshop on positioning, navigation and communication*. IEEE, 2009, pp. 243–251.
- [125] Teemu Roos, Petri Myllymäki, Henry Tirri, Pauli Misikangas, and Juha Sievänen, “A probabilistic approach to wlan user location estimation,” *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.
- [126] Yunye Jin, Wee-Seng Soh, and Wai-Choong Wong, “Indoor localization with channel impulse response based fingerprint and nonparametric regression,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 1120–1127, 2010.

-
- [127] "<http://indoorlocplatform.uji.es/>," (accessed 10-03-2017).
- [128] B Matha, "Propagation data and prediction methods for the planning of indoor radio communication systems and the radio local area networks in the frequency range 900 mhz to 100 ghz," in *ICST Wireless Internet Conf., ITU-R Recommendations, Geneva*, 2001.
- [129] Peter H Westfall, "Kurtosis as peakedness, 1905–2014. rip," *The American Statistician*, vol. 68, no. 3, pp. 191–195, 2014.
- [130] Jeremy Appleyard, Tomas Kocisky, and Phil Blunsom, "Optimizing performance of recurrent neural networks on gpus," *arXiv preprint arXiv :1604.01946*, 2016.
- [131] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 661–670.
- [132] J. Brownlee, "Machine learning mastery. how to control the speed and stability of training neural networks with gradient descent batch size. available online : <https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>," (accessed 05-03-2019).
- [133] Merima Kulin, Carolina Fortuna, Eli De Poorter, Dirk Deschrijver, and Ingrid Moerman, "Data-driven design of intelligent wireless networks : An overview and tutorial," *Sensors*, vol. 16, no. 6, pp. 790, 2016.

Méthodes de localisation de capteurs dans le contexte de l'Internet des Objets

Résumé

Grace à l'émergence croissante de l'Internet des Objets et à l'importance de l'information de position dans ce contexte, la localisation attire de plus en plus d'attention dans la communauté des chercheurs. La localisation en extérieur est assurée par le GPS qui n'est pas adapté aux environnements intérieurs. Plusieurs techniques de localisation en intérieur existent, mais il n'existe pas encore un standard. L'objectif de cette thèse est d'améliorer les techniques de localisation existantes tout en maintenant un niveau de localisation satisfaisant avec une faible complexité de calcul. Afin de surmonter les inconvénients des techniques de localisation existantes, nous avons proposé des approches alternatives. Dans un contexte de communication distribuée, la trilatération a été combinée avec un processus d'optimisation qui vise à compléter la matrice de distances inter nœuds à partir des données partiellement connues en se basant sur des algorithmes d'optimisation avancés. Ainsi une solution de localisation pour une architecture distribuée a été proposée. Pour une architecture centralisée utilisant la technique du fingerprinting contenant les puissances reçues, plusieurs stratégies ont été étudiées. Une étude comparative entre les différentes métriques d'évaluation de similarité a été développée. Cette étude a été suivie par le développement d'un modèle linéaire entre le fingerprint de test et les fingerprints d'une base de données générant une relation mathématique qui relie les puissances du signal reçues par un objet à ses coordonnées. Cela aide à diminuer la complexité de calcul en ligne et ainsi mieux s'adapter à un système temps réel. Enfin, la relation entre les puissances reçues et les coordonnées a été confiée à un réseau de neurones convolutif (CNN) qui traite le problème de localisation comme un problème de classification d'images radio. Les performances de toutes les approches proposées ont été évaluées et discutées. Ces résultats montrent bien l'amélioration des performances des techniques basiques en termes de précision de localisation et de complexité.

Mots clés : La localisation en intérieur, la puissance du signal reçu, la trilatération, le fingerprinting, la complétion de matrice, les réseaux de neurones convolutifs (CNN).

Résumé en anglais

With the growing emergence of the Internet of Things and the importance of position information in this context, localization is attracting more and more attention in the researchers' community. The outdoor location is provided by GPS which is not suitable for indoors environments. Several indoor localization techniques exist, but there is not yet a standard. Existing methods are mainly based on trilateration or fingerprinting. Trilateration is a geometric method that exploits the distances between an object and reference points to locate it. This method only works when we have at least 3 access points detected and is strongly affected by multi paths. In order to overcome these disadvantages, the fingerprinting method compares the fingerprint associated to the object to be located to a fingerprints' database constructed on offline. The estimated position is a combination of the selected training positions. This method is of great interest. However, it requires significant computing and storage capabilities. The aim of this thesis is to improve the existing localization techniques while maintaining a satisfying localization accuracy with low computational complexity. In order to overcome the disadvantages of these two classes of localization techniques, we propose alternative approaches. For trilateration, it has been combined with an optimization process that aims at completing the inter-node distance matrix from partially known data. Advanced optimization algorithms have been used in developing the mathematical equation corresponding to each one. Using this method, we came up with a localization solution for a distributed IoT architecture. As for fingerprinting, we have exploited it to develop localization systems for a centralized IoT architecture. A comparative study between different metrics of similarity evaluation is conducted. This study was followed by the development of a linear model generating a mathematical relation that links the powers of the signal received by an object to its coordinates. This helps to reduce the online complexity of and adapts our system to real time. This is also ensured by the development of a CNN model which deal with the localization problem as radio images classification problem. The performances of all proposed approaches are evaluated and discussed. These results show the improvement of the performances of basic techniques in terms of localization accuracy and complexity.

Keywords: Internet of Things (IoT), indoor localization, RSSI, trilateration, fingerprints, matrix completion, convolutional neural networks (CNN).