



HAL
open science

Étude de l'application des contraintes géométriques et spatiales à la localisation piétonne en intérieur

Mohamed Anis Ghaoui

► **To cite this version:**

Mohamed Anis Ghaoui. Étude de l'application des contraintes géométriques et spatiales à la localisation piétonne en intérieur. Traitement du signal et de l'image [eess.SP]. Université Paris-Saclay, 2023. Français. NNT : 2023UPASG054 . tel-04721185

HAL Id: tel-04721185

<https://theses.hal.science/tel-04721185v1>

Submitted on 4 Oct 2024

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.

Étude de l'application des contraintes géométriques et spatiales à la localisation piétonne en intérieur

Study of the application of geometric and spatial constraints to indoor pedestrian localisation

Thèse de doctorat de l'université Paris-Saclay

Ecole doctorale n°580,
sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : sciences du traitement du signal et des images
Graduate School : informatique et science du numérique
Réfèrent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **SATIE**
(Université Paris-Saclay, ENS Paris-Saclay, CNRS),
sous la direction de **Roger REYNAUD**, Professeur des universités,
et le co-encadrement de **Bastien VINCKE**, Maître de conférences

Thèse soutenue à Paris-Saclay, le 26 septembre 2023, par

Mohamed Anis GHAOUI

Composition du jury

Membres du jury avec voix délibérative

Myriam SERVIÈRES Professeure des universités École Centrale Nantes - Nantes Université	Présidente
Silvère BONNABEL Professeur des universités Mines Paris - Université Paris Sciences et Lettres	Rapporteur/examineur
Nel SAMAMA Professeur des universités Télécom SudParis - Institut polytechnique de Paris	Rapporteur/examineur
Sylvie LE HÉGARAT-MASCLE Professeure des universités Université Paris-Saclay	Examinatrice

Titre : Étude de l'application des contraintes géométriques et spatiales à la localisation piétonne en intérieur

Mots clés : Localisation en intérieur, Fusion de données, Localisation par vision, Intelligence Artificielle

Résumé : En localisation en intérieur, l'usage de signaux proprioceptifs provenant d'une centrale inertielle portée par le piéton permet d'avoir une première estimation de la position et du cap du piéton. Cette estimation devient vite erronée à cause de l'accumulation des erreurs provenant des capteurs bruités.

Cette thèse se décompose en deux parties. Dans la première, l'étude des contraintes spatiales vise à incorporer la carte d'un bâtiment à un système de localisation en intérieur afin de limiter l'erreur d'estimation. Cette partie propose la représentation de la vraisemblance du mouvement humain dans l'espace intérieur sous la forme d'une grille. Cette représentation a pour effet de réduire l'incertitude sur la position et d'augmenter l'intégrité du système.

La deuxième partie de cette thèse concerne l'introduction d'un signal extéroceptif pour corriger l'estimation. Ce signal est une détection d'amer dans une image par un modèle neuronal. Cette partie propose donc la méthodologie d'adaptation du réseau de neurones au problème de localisation en définissant les besoins, les hypothèses, les outils et les méthodes nécessaires pour obtenir ce signal. Enfin, elle propose une gestion probabiliste des cas où le modèle neuronal produit des faux positifs.

Durant les étapes de sa construction, le système proposé est validé par des séries de tests unitaires et d'expériences de localisation menés pour montrer l'impact des propositions de ce manuscrit.

Title : Study of the application of geometric and spatial constraints to indoor pedestrian localisation

Keywords : Indoor localisation, Data fusion, Localisation by vision, Artificial Intelligence

Abstract : In indoor localisation, the use of a proprioceptive signals coming from an inertial measurements unit carried by the pedestrian offers a first estimate of the position and heading of the pedestrian. This estimate quickly becomes erroneous because of the accumulation of errors from noisy sensors data.

This thesis is divided into two parts. In the first, the study of spatial constraints aims to incorporate the map of a building with an indoor localisation system in order to limit the estimation error. This part proposes the representation of human motion likelihood in interior space in the form of a grid. This representation has the effect of reducing the uncertainty on the position and increasing the integrity of the system.

The second part of this thesis concerns the introduction of an exteroceptive signal to correct the estimation. This signal is a detection of landmark in an image by a neural model. This part therefore proposes the methodology for adapting the neural network to the problem of localisation by defining the needs, the hypotheses, the tools and the methods necessary to obtain this signal. Finally, it proposes a probabilistic management of cases where the neural model produces false positives.

During each step of its construction, the proposed system is validated by a series of unitary tests and experiments conducted to demonstrate the impact of the propositions of this thesis.

J'écris ces modestes mots après avoir soutenu ma thèse et je souhaite remercier ceux qui y ont contribué. Ces trois ans m'ont transformé davantage que les vingt-quatre qui les ont précédés. Debout sur les épaules des géants, je réalise l'un de mes rêves les plus fous.

Je tiens d'abord à remercier les rapporteurs, les professeurs *Bonnabel* et *Samama*, d'avoir accepté de juger ma thèse et d'y avoir consacré leur temps et leur expertise. Je tiens aussi à remercier mes examinatrices, les professeures *Le Hégarat-Masclé* et *Servières*, d'avoir accepté de faire partie de mon jury. Enfin, je remercie Dr. Michel d'avoir accepté mon invitation pour assister à ma soutenance et pour les différentes discussions, en ligne ou en personne, que nous avons eues.

Je tiens à remercier l'équipe encadrante composée de *Roger Reynaud* et *Bastien Vincke* pour leur soutiens indéfectible dans la lourde charge de mon encadrement.

Les discussions scientifiques que j'ai eues avec Roger m'ont formé en tant que docteur. Ses réflexions plus philosophiques m'ont marqué et resteront gravées dans ma mémoire. C'est grâce à sa sagesse et à son expérience que j'ai pu mener ce travail à bien. Je me souviens encore du mail où tu as accepté d'être mon directeur de thèse. Merci à toi.

La rigueur et l'exigence de Bastien concernant mes méthodes, mes présentations et mes rédactions ont été exemplaires. Le travail et les échanges avec Bastien m'ont toujours apporté une critique constructive. Je me souviens encore du mail que j'ai reçu de ta part pour mon stage en M1. Merci à toi.

À ce propos, je tiens à remercier *Abdelhafid Elouardi*, qui était mon enseignant en M1 et qui m'a recommandé pour un stage à *Bastien*.

Je salue les administrations de l'école doctorale STIC, du laboratoire et de l'Université Paris-Saclay pour leur soutien et leur aide face aux divers problèmes rencontrés durant ces trois ans.

Mes remerciements s'adressent également au personnel de l'IUT d'Orsay, où j'ai enseigné pendant ces trois années. Ces cent quatre-vingt-douze heures passées avec mes étudiants furent un pur plaisir.

C'est au tour de mes collègues du laboratoire SATIE, qu'ils soient permanents ou non, de recevoir mes remerciements. Les différentes conversations et échanges quotidiens ont fait en sorte que je ne me sois jamais senti isolé au travail. Je remercie tout particulièrement *Emi*, *Pauline* et *Moha*.

Sur une note personnelle, je remercie ma famille et mes amis, qu'ils soient proches ou éloignés, présents ou passés. Quelques citations pour décrire ces trois années passées :

*** *À tous ceux qui y ont cru et surtout à ceux qui n'y ont pas cru.* ***

*** *Nous prenons le risque pour la récompense.* ***

*** *La mort est le sort de tous les guerriers. Un sort que nous pourrions bientôt partager.* ***

*** *Que les forces du mal soient désorientées sur le chemin de votre maison.* ***

*** *Pour Oria, Adam et moi-même.* ***

Table des matières

1	Introduction Générale	17
1.1	Contexte général	18
1.2	Applications	18
1.3	Verrous scientifiques et objectifs	20
1.3.1	Verrous scientifiques	20
1.3.2	Objectifs	21
1.4	Structure du manuscrit	22
1.5	Contributions	23
I	Localisation proprioceptive	25
2	État de l’art des modèles de marche piétonne et de correction de trajectoires	27
2.1	Modèles de marches	28
2.1.1	Modèle biomécanique	29
2.1.2	Intégration avec détection du pas et modèle d’état	29
2.2	Navigation piétonne à l’estime (PDR)	30
2.2.1	Détection de pas	32
2.2.2	Estimation de la longueur du pas	32
2.2.3	Estimation du cap de pas	33
2.3	Localisation par infrastructure réseau	34
2.4	Localisation par vision par ordinateur	36
2.5	Fusion de données pour la localisation	36
2.5.1	Filtrage sur modèle d’état	36
2.5.2	Usage dans la localisation	38
2.6	Synthèse	40
2.7	Structure de la partie I	41
2.7.1	Contributions	42
3	Acquisitions des données, systèmes de coordonnées et tests de validation	43
3.1	Acquisition de données	44
3.1.1	Alignement Spatial	44
3.1.2	Alignement Temporel	45
3.1.3	Calibration des capteurs	45
3.2	Systèmes de coordonnées - SC	47
3.2.1	Système de coordonnées global - SCG	47
3.2.2	Système de coordonnées utilisateur - SCU	47

3.2.3	Système de coordonnées appareil - SCA	48
3.3	Hypothèses prises pour cette thèse	51
3.3.1	Position et orientation de l'appareil par rapport à l'utilisateur	51
3.3.2	Verticalité du piéton	52
3.3.3	Mise à jour événementielle	52
3.4	Tests de validation	52
3.4.1	Test d'acquisition statique	52
3.4.2	Calibration et non-calibration par Android	53
3.4.3	Tests des translations unitaires	54
3.4.4	Tests des rotations unitaires	58
3.5	Récapitulatif	61
3.6	Conclusion	62
4	Localisation par fusion de sources primaires	63
4.1	Navigation piétonne à l'estime (PDR)	64
4.1.1	Pré-traitement des données inertielles	64
4.1.2	Détection de pas	64
4.1.3	Estimation de la longueur de pas	66
4.1.4	Estimation du cap de pas	69
4.2	Métriques d'évaluation	70
4.2.1	<i>EvAAL framework</i>	71
4.2.2	Distances statistiques	72
4.2.3	Fonction de distribution cumulative (CDF)	73
4.2.4	Retour au départ	73
4.3	Vérité Terrain	73
4.4	Expérience : estimation d'une trajectoire	76
4.5	Récapitulatif	77
4.6	Conclusion	78
5	Correction de la trajectoire par fusion de sources secondaires	81
5.1	Fusion par filtre récursif	82
5.1.1	Filtrage bayésien	82
5.1.2	Modèle d'état dans le filtre particulaire PF	84
5.2	Étapes du filtrage à particules	85
5.2.1	Initialisation	85
5.2.2	Prédiction	87
5.2.3	Mise à jour	89
5.2.4	Ré-échantillonnage	90
5.3	Filtrage par murs	92
5.3.1	Formats de carte	92
5.3.2	Contraintes géométriques pour filtre particulaire	95
5.3.3	Régénération	95
5.3.4	Sortie du filtre	96
5.3.5	Sortie du filtre contrainte	97
5.4	Filtrage par modèle de mouvement humain	97
5.4.1	Grille de vraisemblance	98
5.4.2	Fonctions d'appartenance	98

5.5	Expériences	99
5.5.1	Étude de l'effet du nombre de particules	100
5.5.2	Étude de l'effet du gain de Weinberg	101
5.5.3	Trajectoire réalisée	102
5.5.4	Effet de la grille de vraisemblance	104
5.6	Récapitulatif	108
5.7	Conclusion	110

II Localisation assistée par vision 111

6	État de l'art de la vision par caméra en localisation	113
6.1	Caméra fixe	114
6.2	Caméra mobile	115
6.2.1	Caméras 2D vs 3D	115
6.2.2	Amers artificiels vs organiques	116
6.2.3	Localisation et cartographie simultanées (SLAM)	117
6.2.4	Réalités augmentée (AR), virtuelle (VR) et mixte (MR)	120
6.2.5	Traitement traditionnel vs intelligence artificielle	121
6.3	Détecteurs d'objets dans une image	122
6.3.1	YOLO	122
6.3.2	Concurrents de YOLO	125
6.4	Synthèse	126
6.5	Structure de la partie II	129
6.5.1	Contributions	129
7	Détection et apprentissage pour la localisation	131
7.1	YOLOv5	132
7.2	Méthodologie de l'adaptation de YOLO pour recalage visuel	133
7.2.1	Méthodologie	133
7.2.2	Amers	134
7.2.3	Choix des classes d'amers	135
7.2.4	Base de données	136
7.3	Apprentissage par transfert	140
7.3.1	Congélation des couches	141
7.3.2	Entraînement du détecteur	142
7.3.3	Sur-apprentissage et arrêt prématuré	142
7.4	Évaluation	143
7.4.1	Métriques	143
7.4.2	Résultats	145
7.5	Faux Positifs et ré-entraînement de YOLO	147
7.5.1	Types de faux positifs	148
7.5.2	Ré-entraînement	150
7.5.3	Expériences	154
7.6	Récapitulatif	157
7.7	Conclusion	158

8	Correction de trajectoire par observation de signaux extéroceptifs	159
8.1	Recalage visuel	160
8.1.1	Alignement temporel détection-pas	160
8.1.2	Construction de l’observation	161
8.1.3	Phase d’appariement	162
8.1.4	Mise à jour des particules	163
8.1.5	Expériences et résultats	165
8.1.6	Récapitulatif	169
8.2	Gestion des faux positifs	169
8.2.1	Applications des contraintes géométriques de la carte	169
8.2.2	Vote majoritaire et définitions des groupes	171
8.2.3	Expériences et résultats	172
8.2.4	Récapitulatif	178
8.3	Conclusion	179
9	Bilan et conclusion généraux	181
9.1	Perspectives	184
9.1.1	Embarquabilité et encapsulation sur téléphone portable	184
9.1.2	Consommation énergétique	184
9.1.3	Partage de carte dans un cadre multi-agent	184
	Annexes	207
	Annexe A Acquisition de données sur Android	209
	A.1 Disponibilité sur <i>GooglePlay</i>	209
	Annexe B Tests unitaires	211
B.1	Tests unitaires de l’estimation du déplacement	211
B.1.1	Test de détection de pas	211
B.1.2	Test de passage sur les points repères	212
B.1.3	Test de distance parcourue	213
B.1.4	Test de trajectoire rectiligne	213
B.1.5	Test de l’invariance aux utilisateurs	214
B.1.6	Test de trajectoire curviligne	214
B.1.7	Test de trajectoires par navigation à l’estime	215
B.1.8	Test de l’orientation en 3D	216
B.1.9	Test de trajectoire par navigation à l’estime 3D	217
B.2	Tests unitaires du filtrage particulaire	219
B.2.1	Test unitaire du filtrage par murs	219
B.2.2	Trajectoire carrée filtrée et régénérée	219
B.3	Adjacence et norme de distance	221
B.4	Algorithmes de ré-échantillonnage	222
	Annexe C Vision par caméra	225
C.1	Modèle de caméra sténopé	225
C.2	Validation par ratio géométrique de la détection	229
C.3	Apprentissage et YOLO	230

C.4 Décomptes des classes dans COCO	232
---	-----

Table des figures

2.1	Modalités de la localisation piétonne en intérieur	28
2.2	Étapes du cycle de marche	29
2.3	Schéma représentant le principe de la navigation piétonne à l'estime	30
3.1	Changement du SC Utilisateur à SC Global.	48
3.2	Transformations de SCA à SCG.	49
3.3	Relation entre les différents Système de Coordonnées.s	50
3.4	Le piéton et les SCs.	51
3.5	Visualisation de la séquence statique de l'accélération.	53
3.6	Visualisation de la séquence statique de la vitesse angulaire.	53
3.7	Visualisation des mesures gyroscopiques calibrées par Android et non-calibrées.	54
3.8	Visualisation des données du test de translation sur l'axe x.	55
3.9	Visualisation de la norme d'accélération du tests de translation sur l'axe x.	56
3.10	Visualisation des données du test de translation sur l'axe y.	56
3.11	Visualisation de la norme d'accélération du tests de translation sur l'axe y.	57
3.12	Visualisation des données du test de translation sur l'axe z.	58
3.13	Visualisation de la norme d'accélération du tests de translation sur l'axe z.	58
3.14	Visualisation de la séquence de rotation autour de l'axe x du gyroscope.	59
3.15	Visualisation de la séquence de rotation autour de l'axe y du gyroscope.	60
3.16	Visualisation de la séquence de rotation autour de l'axe z du gyroscope.	60
4.1	Accélérations filtrées et non filtrées. Fréquence de coupure = 2 Hz.	65
4.2	Détection d'un pas visualisée.	65
4.3	Schéma représentant un pas	67
4.4	Visualisation des 3 axes du gyroscope d'un téléphone.	69
4.5	Comparaison entre l'évolution des angles ϕ , ψ et α	70
4.6	Utilisateur déclarant son passage sur un point repère.	76
4.7	Trajectoire estimée par PDR.	77
4.8	Schéma bloc des fonctions présentées.	77
5.1	Schéma récapitulatif du système actuel pour un pas.	82
5.2	Schéma de positionnement du filtrage à particules	84
5.3	Initialisation globale sur la carte avec 20000 particules.	86
5.4	Initialisation de 2000 particules avec les bruits δ_0 et α_0	87
5.5	Comparaison entre prédictions déterministe et probabiliste pour 5 particules.	88
5.6	Prédictions du déplacement avec 10^6 particules.	89
5.7	Phénomène de dégénérescence visualisé.	90

5.8	Représentation d'un espace en intérieur par un graphe.	94
5.9	En vert, l'ensemble de \widehat{N} particules sélectionnées pour la régénération.	96
5.10	Particules distribuées autour d'un virage.	97
5.11	Concepts de grille.	99
5.12	Grille de vraisemblance construite à partir du plan de la carte	100
5.13	Évolution des erreurs mesurées en fonction du nombre de particules	101
5.14	Évolution des distances mesurées en fonction du gain de Weinberg.	102
5.15	Trajectoire d'un piéton estimée par le PF.	103
5.16	Représentation de toutes les particules de la trajectoire de la figure 5.15.	103
5.17	Représentation de toutes les particules de la trajectoire.	104
5.18	Ellipsoïdes d'incertitude visualisés sans la grille de vraisemblance.	105
5.19	Ellipsoïdes d'incertitude visualisés avec la grille de vraisemblance.	105
5.20	Erreurs signées mesurées $\pm 3\sigma$ à chaque point repère.	106
5.21	Comparaison entre les estimations du PF, RoNIN et PDR.	107
6.1	Schéma récapitulatif de <i>Kimera</i>	119
6.2	Diagramme récapitulatif de YOLOv5	123
6.3	Résultats d'évaluation des différentes tailles de YOLO	125
7.1	Positionnement de notre proposition de méthodologie.	133
7.2	Étiquetage de deux extincteurs.	137
7.3	Autres cas d'extincteurs.	137
7.4	Échantillon d'entraînement augmenté par mosaïque.	138
7.5	Corrélogrammes des étiquettes.	139
7.6	Répartition des étiquettes.	140
7.7	Procédure d'entraînement par transfert.	141
7.8	Entraînement de YOLO basé sur le diagramme de la figure 6.2.	142
7.9	Visualisation de l'intersection (rouge) sur l'union (magenta).	144
7.10	Courbes de performances (1)	146
7.11	Courbes de performances (2)	147
7.12	Courbes d'évolution des métrique de l'entraînement sur 300 époques.	148
7.13	Plusieurs exemples de faux positifs.	149
7.14	D'autres exemples de faux positifs.	150
7.15	D'autres exemples de faux positifs.	150
7.16	Exemple de détections à travers des murs "transparents".	151
7.17	Images contenant le même extincteur prises à différentes profondeurs de champ.	151
7.18	Répartition des étiquettes pour les deux classes : humain et extincteur.	152
7.19	Courbes d'évolution des métrique de l'entraînement sur 200 époques.	153
7.20	Courbes de performances.	153
7.21	Confusion entre extincteur et humain.	154
7.22	Humain et extincteur distingués.	155
7.23	Réflexions présentes dans l'environnement.	155
7.24	Détections de plusieurs humains à différentes échelles.	156
7.25	Détection sur un faux positif fabriqué.	156
8.1	Système proposé dans ce chapitre.	160
8.2	Chronogramme des événements.	161

8.3	Système de coordonnées de particule (UCS) par rapport au système de coordonnées global (GCS).	162
8.4	Représentation de plusieurs particules devant s'apparier sur un amer L	163
8.5	Résultat du processus d'appariement observation - amer.	164
8.6	Trajectoires pour quatre modes différents.	165
8.7	Comparaison des erreurs euclidiennes et Mahalanobis moyennes.	166
8.8	Comparaison des scores CDF pour les quatre modes.	167
8.9	Comparaison des critères d'intégrité pour les quatre modes.	167
8.10	Comparaison des critères d'intégrité pour le mode rouge pour différentes σ	168
8.11	Comparaison de la présence de grille sur le gain de <i>Weinberg</i> avec recalage visuel toujours actif.	168
8.12	Visualisation de la validation par largeur.	170
8.13	Schéma bloc de l'intégration du processus de validation.	171
8.14	Visualisation 3D des particules.	172
8.15	Regroupement des particules sur une trajectoire.	173
8.16	Regroupement des particules à un pas donné pour λ , 2λ et 3λ	173
8.17	Comparatif entre deux traitements pour le groupe non validant.	174
8.18	Trajectoire 1.	175
8.19	Trajectoire 2.	175
8.20	Trajectoire 3.	176
8.21	Trajectoire 4.	177
8.22	Trajectoire 5.	177
A.1	Résultat de la recherche du mot clef "sensors" sur <i>GooglePlay</i>	209
B.1	Visualisation de la détection de pas sur la norme d'accélération.	211
B.2	Sept pas effectuées sur points repères.	212
B.3	Trajectoire rectiligne visualisée.	213
B.4	Trajectoire curviligne visualisée.	215
B.5	Trajectoire aller-retour visualisée.	215
B.6	Trajectoire carrée visualisée.	216
B.7	Évolution des angles d'orientation prédits par le filtre de Madgwick.	217
B.8	Système de coordonnées sphériques.	218
B.9	Visualisation de trajectoires en 3D.	219
B.10	Filtrage des particules par les murs.	219
B.11	Propagation de 50 particules visualisée sur une trajectoire carrée avec :	220
B.12	Visualisation de 5 algorithmes de ré-échantillonnage	222
C.1	Vue à vol d'oiseau des systèmes de coordonnées de la figure 8.3.	225
C.2	Damier de calibration.	226
C.3	Motif de damier.	227
C.4	Disposition du télémètre laser sur le repère au sol face à l'extincteur.	228
C.5	Fonction d'étalonnage : $du = g(distance)$	229
C.6	Statistiques des hauteur/largeur de détections du jeu d'entraînement.	230
C.7	Architecture de YOLOv5 en sous blocs élémentaires	230

Liste des tableaux

2.1	Résultat de la précision de classification du mode de déplacement	32
2.2	Tableau récapitulatif des technologies de localisation dépendante de l'espace en intérieur	36
3.1	Liste des applications d'acquisition de données retenues puis testées.	44
3.2	Tableau comparatif de l'estimation de l'orientation d'un téléphone	47
3.3	Biais des mesures gyroscopiques calibrées et non-calibrées.	54
4.1	Résultats de la comparaison des estimation de distances	68
5.1	Paramètres du filtre particulaire pour la figure 5.6b.	88
5.2	Gains de <i>Weinberg</i> empirique et statistique.	101
5.3	Résultats numériques de la comparaison de figure 5.21	107
6.1	Tableau de comparaison en précision et temps d'exécution	124
7.1	Liste de classes présentent dans FireNet	136
8.1	Distances moyennes pour les cinq trajectoires.	178
B.1	Données prélevées et calculées sur différents utilisateurs.	214
B.2	Moyennes et écart-types des angles ψ et ϕ pour la figure B.7.	217
C.1	Lists de catégories et leurs classes dans COCO <i>train2017</i>	232

Chapitre 1

Introduction Générale

Sommaire

1.1	Contexte général	18
1.2	Applications	18
1.3	Verrous scientifiques et objectifs	20
1.3.1	Verrous scientifiques	20
1.3.2	Objectifs	21
1.4	Structure du manuscrit	22
1.5	Contributions	23

1.1 Contexte général

La localisation en intérieur est une technologie qui vise à déterminer avec précision la position d'un objet ou d'une personne à l'intérieur de bâtiments ou d'espaces clos. Alors que la localisation en extérieur est bien établie grâce à l'utilisation du GNSS, la localisation en intérieur pose des défis uniques en raison de l'absence de signaux GNSS fiables à l'intérieur des bâtiments [Zafari et al., 2019, Gu et al., 2019]. La demande croissante de services basés sur la localisation en intérieur, tels que la navigation dans les centres commerciaux, les musées, les aéroports, les hôpitaux ou même les entrepôts, a stimulé la recherche et le développement de solutions innovantes. Les avancées technologiques récentes, combinées à l'émergence de l'Internet des objets (IoT) et de la connectivité sans fil, ont ouvert de nouvelles possibilités pour la localisation en intérieur.

Plusieurs technologies sont utilisées pour la localisation en intérieur, notamment le WiFi, le Bluetooth, les capteurs inertiels, les balises d'identification par radio-fréquence (RFID), les systèmes de caméras et bien d'autres encore. Ces technologies peuvent être utilisées de manière indépendante ou combinées pour obtenir une précision plus importante. Cependant, la localisation en intérieur soulève également des questions de confidentialité et de sécurité, car elle nécessite la collecte de données sensibles sur les individus. Il est donc essentiel de mettre en place des mesures de protection des données et de garantir le consentement et la transparence pour prévenir les abus.

Dans le cas de l'humain, la marche est le mode principal de déplacement. De plus, le piéton passe 90% de son temps dans des espaces en intérieur. Il est donc compréhensible que le besoin de se localiser pour naviguer et se diriger soit si important. La localisation en intérieur des piétons est donc un domaine de recherche en évolution rapide, avec des avancées récentes dans les technologies utilisées pour obtenir la position et/ou l'orientation du piéton dans le bâtiment.

Cette thèse étudie la localisation du piéton pendant sa marche à l'intérieur du bâtiment. Elle vise à proposer un système de localisation en exploitant des processus déjà existants, en les validant et en les intégrant comme une brique élémentaire. L'étude de chacune de ses briques est présentée dans ce document qui est composé de deux parties : la première partie traite de la localisation purement proprioceptive. Alors que la deuxième partie présente l'usage d'un signal extéroceptif via la caméra. La carte du bâtiment est considérée comme une source d'information supplémentaire qui permet de compléter l'information qu'elle soit proprioceptive ou extéroceptive. Dans les deux parties, les contraintes géométriques et spatiales tirées de cette carte sont déployées afin d'assister les différents processus.

1.2 Applications

Les applications de localisation intérieure deviennent de plus en plus populaires en raison de leur capacité à localiser avec une précision satisfaisante l'emplacement d'un piéton dans un bâtiment. Ces applications comprennent plusieurs champs de l'industrie, de la vie urbaine et domestique.

Les robots sont de plus en plus utilisés dans des lieux intérieurs, tels que des installations industrielles, des entrepôts et des hôpitaux. La localisation peut aider les robots à naviguer dans ces espaces intérieurs. Les robots peuvent être programmés avec une carte de l'espace

intérieur, et le système de localisation intérieure peut être utilisé pour aider les robots à déterminer leur position et orientation dans l'espace. Cela peut être utile pour des tâches telles que la livraison d'articles à des endroits spécifiques, l'inspection de zones de l'espace et l'évitement d'obstacles.

Une application concerne le commerce où les commerçants peuvent utiliser la technologie pour mieux comprendre leurs clients [Kamiya et al., 2019, Renaudin, 2019]. Par exemple, un centre commercial peut utiliser la technologie pour suivre les déplacements des clients dans le centre et identifier les zones les plus populaires. Ces informations peuvent ensuite être utilisées pour améliorer l'expérience client en plaçant stratégiquement des éléments tels que des distributeurs de billets, toilettes, panneaux d'information, ...

Elles peuvent aussi être utilisées dans le secteur de la santé et de la sécurité. Par exemple, les hôpitaux et autres établissements médicaux peuvent utiliser un service de localisation pour améliorer la sécurité des patients en s'assurant que le personnel et l'équipement médical se trouvent au bon endroit au bon moment [Shipkovenski et al., 2020, Gholamhosseini et al., 2019]. De plus, la technologie peut être utilisée pour suivre l'équipement médical afin de s'assurer qu'il n'est pas égaré ou volé.

La sécurité des personnes dans les espaces intérieurs peut être assurée par un système de localisation. Ce système peut être utilisé pour suivre l'emplacement des personnes dans l'espace, afin que si quelqu'un est perdu ou blessé, il puisse être rapidement localisé et aidé. Le système peut également être utilisé pour garder une trace de l'emplacement des matières dangereuses ou des zones qui doivent être évitées.

Un cas plus avancé est celui du déploiement d'une équipe de pompiers ou d'ambulanciers qui en utilisant un système de localisation pour le bâtiment cible, peut rapidement naviguer à l'intérieur, connaître la position des membres de l'équipe, signaler la présence d'un danger en mettant à jour la carte de localisation ainsi qu'avoir des informations contextuelles sur le lieu, par exemple, la présence d'un conduit de gaz caché à proximité.

Les systèmes de guidage des piétons sont de plus en plus populaires dans les espaces intérieurs, tels que les centres commerciaux, hôpitaux, aéroports, ... ces systèmes peuvent être utilisés pour aider à guider les piétons vers l'emplacement souhaité. Le système de localisation intérieure peut suivre l'emplacement du piéton, de sorte qu'il puisse recevoir des indications précises vers sa destination. Cette application sert aussi à guider les personnes aveugles et malvoyantes [Cheraghi et al., 2021, Fusco et al., 2020, Kunhoth et al., 2020].

Les foules sont des entités répondant à des comportements de groupes qui agissent sur chaque individu et peuvent parfois représenter un danger pour ce dernier. Les systèmes de localisation intérieure peuvent également être utilisés pour gérer les foules dans les espaces intérieurs, tels que les stades, le métro parisien et le pèlerinage autour de la Mecque. Le système de localisation intérieure peut suivre l'emplacement des personnes dans l'espace, afin que la foule puisse être gérée et dispersée si nécessaire. Cela peut aider à assurer la sécurité de tout le monde dans l'espace, ainsi qu'à réduire la congestion.

Les applications de Réalité Virtuelle (VR) et de Réalité Augmentée (AR) sont de plus en plus utilisées dans les espaces intérieurs. Les systèmes de localisation en intérieur sont essentiels pour que les expériences VR/AR soient précises et immersives. Le système de localisation intérieure est utilisé pour suivre l'emplacement de l'utilisateur dans l'espace, de sorte que l'expérience VR/AR puisse être adaptée à son emplacement exact.

Il est aussi possible d'améliorer l'expérience cinématographique dans les cinémas en salle. Le système de localisation intérieure peut être utilisé pour suivre l'emplacement du spectateur

dans le cinéma, afin que le film puisse être adapté à son emplacement exact. Par exemple, le son peut être réglé de manière à ce que le spectateur entende toujours le son provenant de la bonne direction.

Enfin, en cas d'urgence, un système de localisation intérieure peut être utilisé pour aider les personnes à évacuer un bâtiment rapidement et en toute sécurité. Le système de localisation intérieure peut être utilisé pour suivre l'emplacement des personnes dans le bâtiment, afin qu'elles puissent être rapidement localisées et guidées vers la sécurité.

Dans l'ensemble, les applications de localisation en intérieur offrent une variété d'avantages et peuvent être utilisées dans un large éventail d'industries. Au fur et à mesure que ces applications deviennent plus populaires, leurs applications potentielles ne feront qu'augmenter.

1.3 Verrous scientifiques et objectifs

1.3.1 Verrous scientifiques

Toutes les applications citées précédemment utilisent différentes modalités afin de répondre à la problématique de localisation. Dans le chapitre 2 de ce manuscrit, l'état de l'art de la localisation présente les verrous scientifiques suivants :

1. Les données inertielles proprioceptives provenant d'un téléphone porté par l'utilisateur sont fiables mais bruitées. L'exploitation de ces données fait que l'estimation du cap est vouée à dériver en l'absence des données magnétométriques exploitables. Donc, il est nécessaire de corriger cette dérive en introduisant une information supplémentaire.
2. Les méthodes d'estimation de distance parcourue dans le cas d'une navigation piétonne à l'estime connaissent les mêmes imprécisions lors de l'estimation à cause des données d'accélération bruitées. Si l'erreur peut être négligeable sur un seul pas, le cumul d'erreurs fait que la trajectoire devient incohérente sur le long terme.
3. Dans le cas où un système probabiliste est utilisé pour l'estimation de la position et du cap du piéton, l'incertitude de ce système est vouée à croître tout au long de la trajectoire par la combinaison des deux effets précédents. Il est nécessaire de limiter cette incertitude afin de proposer un système qui garde la position correcte dans un voisinage acceptable.
4. Dans le cas où le système de localisation utilise l'infrastructure réseau du bâtiment, la consommation énergétique par les équipements de ce réseau est en augmentation drastique. Les projections de consommation énergétique sont importantes, voir inquiétante et impossible à atteindre si la localisation est un service omniprésent dans les bâtiments. Les systèmes de localisation conçus doivent tenir compte de cette consommation énergétique, surtout dans le cas où la cible de déploiement est le téléphone portable.
5. Le pistage des personnes est une pratique qui connaît une très grande réticence lors de l'adoption d'un service tel que la localisation. La confidentialité sur la position de l'individu est un enjeu primaire. Un système de localisation doit répondre à cette question de confidentialité en éclaircissant sa gestion de cette information et en présentant de manière transparente l'usage de celle-ci. Dans le cas du téléphone portable, il est intuitif de vouloir encapsuler le système déployé dedans afin de garantir le respect de la confidentialité lors du positionnement.

6. Les réseaux de neurones sont des modèles d'inférence d'information à partir de données primaires. Ces modèles ont connu une grande vague d'intérêt dans la dernière décennies pour leur performances inégalées et commencent à être omniprésents. L'usage d'un de ses modèles dans un cas applicatif tel que la localisation présente des difficultés d'adaptation et des spécificités dont il faut tenir compte.
7. Quand un système de localisation se repose sur les détections d'objets produites par un réseau de neurones en tant que signal extéroceptif pour corriger ses position et cap, il devient sensible aux faux positifs que ce modèle neuronal peut produire. Si ces faux positifs ne sont pas traités, la localisation peut être fortement dégradée ou même échouer. L'explicabilité des faux positifs provenant de ce modèle neuronal est en elle même une problématique de l'intelligence artificielle. Il revient au système de localisation de proposer un artifice pour tenir compte de la possibilité d'une fausse alarme.

1.3.2 Objectifs

Les objectifs de cette thèse sont les suivants :

[A] **Correction de la position** par le déploiement d'un filtre particulière pour estimer et suivre la position du piéton dans le temps. Ce filtre particulière incorpore les contraintes spatiales qui encadrent le mouvement du piéton.

[B] **Correction du cap** par l'introduction de contraintes spatiales extraites du plan de l'étage dans lequel le piéton est. Ceci est nécessaire car le magnétomètre ne peut pas fournir le cap par rapport au nord magnétique dans un bâtiment dû aux interférences ferromagnétique.

[C] **Réduction de l'incertitude de positionnement** par la modélisation de la vraisemblance du mouvement humain dans des espaces intérieurs sous la forme d'une grille. Cette grille indique aux particules du filtre à quel point la position qu'elles ont prédite est vraisemblable.

[D] **Proposition d'une méthodologie d'adaptation d'un détecteur d'objet** basé sur réseau de neurones pour le recalage visuel sur amer à identifier. Cette méthodologie décrit comment répondre au problème de localisation en important des modèles d'intelligence artificielle tout en tenant compte des spécificités liée à la localisation piétonne en intérieur.

[E] **Renforcement de la localisation en intérieur** par extraction des contraintes géométriques et spatiales de la carte du bâtiment pour l'élimination de faux positifs produits par le détecteur d'objet à base de réseau de neurones.

[F] **Déploiement sur téléphone** en respectant les contraintes mémoires et de temps d'exécution. Le but est de proposer un système dont l'employabilité dans un téléphone soit déjà faisable sans exiger de capteurs supplémentaires ni de changement dans l'architecture des téléphones actuels.

[G] **Sobriété énergétique et confidentialité** en limitant le nombre de capteurs exploités à ceux déjà existants dans le téléphone et en évitant l’usage de l’infrastructure réseau. Le non usage du réseau implique que le système proposé par ce manuscrit est totalement encapsulé. Le système ne dépend d’aucune information du réseau de communication. Il peut donc garantir un fonctionnement confidentiel.

[H] **Incorporation de comportement piéton et localisation multi agents** dans le processus de localisation. Le déplacement piéton peut être conditionné par l’environnement intérieur comme des murs, des meubles, des indications ou d’autres piétons. Le système modélise le comportement du piéton qui le porte et peut connaître la position des piétons autour pour prédire l’évolution de leur déplacement.

1.4 Structure du manuscrit

Cette thèse concerne l’étude de l’exploitation d’un élément intrinsèquement lié à la localisation en intérieur : la carte. Contrairement aux sources inertielles, celle-ci ne provient pas du téléphone. Mais, dans ce manuscrit, son usage permet de la considérer comme une source proprioceptive du fait qu’elle ne génère pas d’événements la mettant à jour. Ce manuscrit est divisé en deux parties en fonction de si les données sont proprioceptives ou extéroceptives. La carte, de par sa nature et son usage peut être considérée comme état l’un ou l’autre selon le contexte.

La première partie concerne les travaux autour de la localisation piétonne en intérieur à base de donnée inertielle. Dans le chapitre 2, l’état de l’art autour des modèles de marche piétonne est exposé. Ensuite, la correction de trajectoire estimée par ces modèles utilisant les systèmes de communication, la vision par ordinateur ou les cartes d’un bâtiment est présentée.

Le chapitre 3 présente la plateforme d’acquisition de données et les systèmes de coordonnées. Le chapitre expose les différents tests de validation menés pour valider l’exploitabilité des données.

Le chapitre 4 explique le processus de navigation piétonne à l’estime (PDR) qui combinent les données acquises en information sur les déplacements du piéton. Ce chapitre présente aussi les solutions pour l’établissement d’une vérité terrain sur les déplacements du piéton. Le chapitre finit par une série de tests validant le fonctionnement de la PDR.

Le chapitre 5 est centré autour de la fusion d’information de sources secondaires dans un formalisme de filtrage particulière. Ce filtre permet de corriger la trajectoire en incluant les contraintes spatiales extraites de la carte. Il présente aussi la représentation de la vraisemblance du mouvement humain dans l’espace intérieur sous la forme d’une grille.

La deuxième partie est axée sur l’usage de la caméra pour assister le processus de localisation. Dans le chapitre 6, un état de l’art des applications de la caméra à la localisation est présenté. Il y est mentionné les cas où les caméras sont statiques ou mobiles. Les localisation et cartographie simultanées (SLAM) et Réalité Augmentée (AR) sont aussi incluses car leurs systèmes utilisent au moins une caméra pour localiser un piéton dans un espace. Enfin, une section de ce chapitre s’intéresse à l’usage de l’intelligence artificielle comme outils pour accomplir cette localisation aidée par la vision.

Le chapitre 7 aborde la méthodologie d’adaptation d’un réseau de neurones convolutif pour produire les détections d’amers nécessaires au recalage visuel. Cette méthodologie justifie les

choix pris en ce qui concerne le modèle neuronal, les données utilisées, l’amer à détecter et la méthode d’entraînement. Dans ce chapitre une campagne de tests est menée pour valider le fonctionnement du réseau de neurones.

Le chapitre 8 met en avant la mise en œuvre du recalage visuel pour le filtre particulaire en exploitant les détections du modèle neuronal. Une gestion des faux positifs par l’incorporation de contraintes géométriques tirées de la carte du bâtiment est aussi présentée dans ce chapitre afin de permettre au système cette gestion probabiliste par groupes de particules.

Le chapitre 9 présente le récapitulatif sur chacune des parties 1 et 2 ainsi qu’une vue d’ensemble plus générale sur cette thèse. Il conclut le manuscrit et présente les perspectives de cette thèse.

1.5 Contributions

Les contributions de cette thèse se répartissent sur chacune des deux parties. Pour la première partie, notre contribution la plus importante publiée dans [Ghaoui et al., 2022] est la modélisation de la vraisemblance du mouvement humain dans les espaces intérieurs sous la forme d’une grille construite à partir de la carte du bâtiment. Cette grille permet d’assigner un poids aux particules du PF ce qui permet de représenter la vraisemblance qu’une particule occupe une position. La grille de vraisemblance permet de resserrer les ellipsoïdes d’incertitude. Dans la finalité, cette grille de vraisemblance peut nous permettre de décrire des comportements de déplacement pour le piéton dans l’espace intérieur.

Notre deuxième contribution est la méthode d’évaluation du système proposé en termes de précision, robustesse et intégrité. Dans [Ghaoui et al., 2022], il est démontré que cette méthode d’évaluation a une vérité terrain satisfaisante en résolution spatiale et temporelle. Les codes et base de données liés à nos expériences de cette partie ont été publiés avec [Ghaoui et al., 2022] et sont disponibles sur <https://github.com/anisghaoui/humolire>.

Dans la deuxième partie de ce manuscrit, notre première contribution est la proposition d’une méthodologie pour l’adaptation d’un détecteur d’objets basé sur des réseaux de neurones convolutifs pour le recalage visuel d’un système proprioceptif sur amers à identifier. Le but de cette méthodologie est d’aider l’exploitation d’un point de vue utilisateur avec une approche boîte noire via un entraînement par transfert pour la mise en œuvre d’une correction de trajectoire par recalage visuel.

Notre deuxième contribution est la validation des détections par l’application de contraintes géométriques tirées de la carte du bâtiment qui permet la possibilité de gérer un faux positif produit par le réseau de neurones via des groupes dont le traitement est différencié [Ghaoui et al., 2023].

Première partie
Localisation proprioceptive

Chapitre 2

État de l’art des modèles de marche piétonne et de correction de trajectoires

Sommaire

2.1	Modèles de marches	28
2.1.1	Modèle biomécanique	29
2.1.2	Intégration avec détection du pas et modèle d’état	29
2.2	Navigation piétonne à l’estime (PDR)	30
2.2.1	Détection de pas	32
2.2.2	Estimation de la longueur du pas	32
2.2.3	Estimation du cap de pas	33
2.3	Localisation par infrastructure réseau	34
2.4	Localisation par vision par ordinateur	36
2.5	Fusion de données pour la localisation	36
2.5.1	Filtrage sur modèle d’état	36
2.5.2	Usage dans la localisation	38
2.6	Synthèse	40
2.7	Structure de la partie I	41
2.7.1	Contributions	42

La localisation en intérieur a connu plusieurs avancées majeures et s’est diversifiée au point de développer des sous-domaines assez distincts. Une manière de les différencier est selon les modalités qu’ils utilisent [Roy and Chowdhury, 2022]. Dans la figure 2.1, les auteurs distinguent trois grands sous-domaines, chacun ayant une approche différente du même problème. Même si ces approches sont représentées comme étant distinctes, les systèmes modernes dans la recherche et l’industrie ont systématiquement une approche qui hybride au moins d’eux d’entre eux [Yassin et al., 2017, Roy and Chowdhury, 2022, Gu et al., 2019].

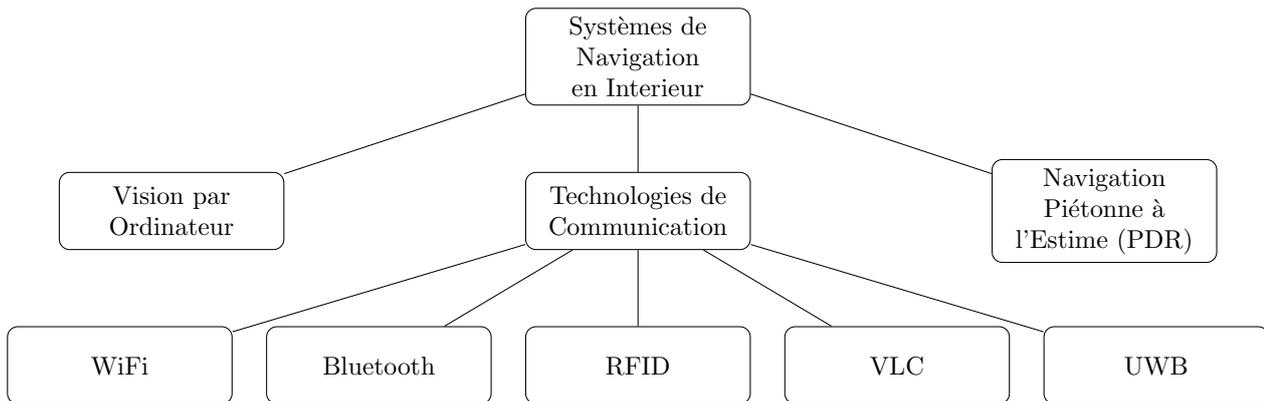


FIGURE 2.1 – Modalités de la localisation piétonne en intérieur adaptée de [Roy and Chowdhury, 2022].

Dans ce chapitre, l’état de l’art autour des systèmes de localisation utilisant la PDR assistée ou non par une modalité complémentaire de vision par ordinateur, de technologie de communication, ou d’utilisation de carte, est exploré. Le but est de pouvoir positionner notre système dans l’état de l’art.

2.1 Modèles de marches

La marche est un mode de déplacement universel pour les êtres humains. Elle est acquise durant l’enfance et est conservée tout au long de la vie. Comprendre la marche est essentiel pour la localisation piétonne, car les mouvements de marche peuvent fournir les informations nécessaires pour prédire la position et/ou le cap d’un piéton.

Un modèle de marche est l’ensemble d’équations mathématiques servant à décrire l’humain durant cette marche. Dans la recherche en localisation piétonne, plusieurs modèles ont été proposés pour représenter la marche [Baker et al., 2017, Kainz et al., 2017, Leboeuf et al., 2019] où le cycle de marche est découpé en plusieurs phases. Dans la figure 2.2, [Stöckel et al., 2015] distingue deux phases elles-mêmes composées de plusieurs étapes : double support où le piéton a ses deux jambes écartées et utilise les deux pour garder son équilibre ; simple support où tout le corps tient en équilibre sur une seule jambe. Et en même temps, l’autre jambe pivote vers la prochaine position.

C’est donc en enchaînant ces deux phases que le piéton peut effectuer un déplacement. Il existe donc une notion de périodicité dans le mouvement qui est aussi présente dans les signaux associés à ceux-ci. Les différents modèles de marche tiennent compte de ces spécificités afin de représenter fidèlement le déplacement du piéton.

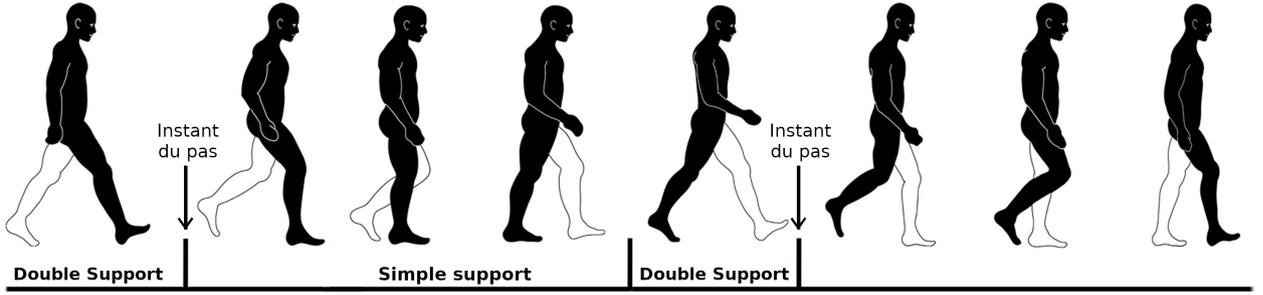


FIGURE 2.2 – Étapes du cycle de marche adaptée de [Stöckel et al., 2015].

2.1.1 Modèle biomécanique

Parmi ces modèles, on trouve les modèles biomécaniques, qui cherchent à représenter les articulations humaines sous la forme d'angles liés à des segments rigides [Baker et al., 2017]. Ainsi, tout mouvement est une combinaison de rotation d'articulations. La recherche biomédicale, la capture de mouvement pour le cinéma et l'identification des piétons exploitent des modèles biomécaniques. Par exemple, l'article [Ariyanto and Nixon, 2011] utilise un modèle biomécanique pour extraire les caractéristiques permettant d'identifier un piéton. Cela est réalisé en représentant le piéton sous la forme de cylindres 3D joints et articulés dans une salle équipée d'un système de capture de mouvement [Vicon, 2023].

Les industriels proposent différentes solutions pour simuler les piétons. Ces solutions se distinguent en termes de finesse/rapidité de simulation, de capacité à gérer les interactions entre piétons, d'inférence de comportement individuel et de groupes, de planification de trajectoires, d'échelle de déploiement, de la fidélité de reproduction des scénarios, ... [Kleinmeier et al., 2019, Bentley - Legion Simulator, 2023, Mass motion, 2023, Oasys, 2023, PTV Viswalk, 2023]. Une liste exhaustive est proposée par [Richards, 2020] pour les classifier selon leur statut : logiciel libre ou propriétaire, et leur capacités.

Cependant, ces modèles sont complexes et leur utilisation pour la localisation piétonne est limitée en raison de leur complexité et de la zone de couverture limitée des dispositifs utilisant ces modèles [Li et al., 2018].

2.1.2 Intégration avec détection du pas et modèle d'état

Une autre catégorie de modèles choisit de représenter le piéton à un instant t par un état E_t . E_t peut être représenté par l'équation (2.1) avec $[x, y]$ la position et θ le cap du piéton.

$$E_t = \{[x_t, \quad y_t] \quad , \quad \theta_t\} \quad (2.1)$$

Ensuite, ces modèles mettent à jour cet état en fonction de l'évolution dans son déplacement. Cette évolution est modélisée par l'équation (2.2) où \mathcal{M} est le modèle de marche. et dans le cas d'une localisation 2D, le déplacement est défini par deux composantes : longueur du pas SL et différence de cap $d\theta$. Il y a donc une représentation discrète du piéton qui est modifiée par des événements discrets déduits de sa marche.

$$E_{t+1} = \mathcal{M}(E_t, \quad \{SL_t, \quad d\theta_t\}) \quad (2.2)$$

Dans le cas d'une modélisation par modèle d'état, il s'agit alors d'une navigation piétonne à l'estime (PDR). La bibliographie autour de la PDR est présentée en section 2.2. Les modèles de marches liés à cette technique sont présentés dans la section 4.1.2.

2.2 Navigation piétonne à l'estime (PDR)

Il s'agit d'une méthode pour estimer le mouvement humain qui repose sur l'idée que la position d'un piéton peut être déterminée en mesurant son déplacement à chaque pas et en combinant cette distance à sa position précédente. Pour ce faire, la méthode implique une détection des pas, une estimation de leur longueur et direction, suivie du calcul de la nouvelle position. Ainsi, le déplacement est discret et consiste en une série de positions que le piéton prend à chaque pas [Zafari et al., 2019]. Afin de correspondre avec nos objectifs [F & G], dans le cas où une Unité de Mesures Inertielles (IMU) est utilisée par la PDR, cette IMU est embarqué dans un téléphone comme dans la figure 2.3. Les valeurs des positions (x, y) sont considérées relatives à la position initiale. Le cap θ est orienté suivant \vec{OX} et est relatif au cap initial. Alors que les valeurs SL et $d\theta$ sont relatives par rapport au pas précédent.

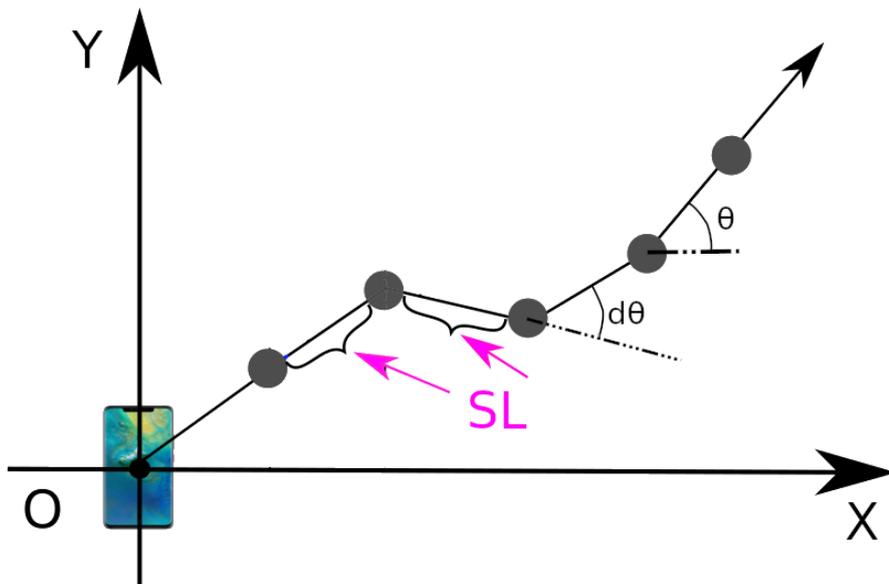


FIGURE 2.3 – Schéma représentant le principe de la PDR adapté de [Gu et al., 2019].

Les techniques reposant sur la compréhension du mouvement humain doivent être flexibles afin de s'adapter aux différents schémas de marche des individus. Une telle adaptabilité est loin d'être atteinte, et donc les modèles de mouvement sont approximés [Murata et al., 2019]. Il est souvent mentionné que les humains ont plusieurs modèles de mouvement parmi : avancer, reculer, esquiver (pas de côté), tourner, ... et chacun d'eux présente des signatures de capteurs spécifiques. C'est pourquoi la reconnaissance du mouvement humain est problématique en soi [Zhang et al., 2019, Lin et al., 2020]. La reconnaissance de mouvement peut être faite par l'analyse des caractéristiques des signaux telles que les variances d'accélération et de vitesse angulaires, les énergies de ses signaux, leur répartition sur le spectre des fréquences de marches... [Susi et al., 2013].

La seconde problématique est celle du placement de l'IMU sur le corps humain. Ce placement a un impact important sur les techniques qui peuvent être appliquées aux signaux collectés. Car différentes configurations : *texting, swinging, pocket, bag,...* peuvent permettre d'émettre différentes hypothèses sur le mouvement effectués [Klein et al., 2018].

Enfin, une partie de la recherche tente de répondre à ces problématiques via l'apprentissage qu'il soit machine ou profond ("*machine or deep learning*"). [Yan et al., 2019, Chen et al., 2018, Gu et al., 2019, Klein and Asraf, 2020] ont proposé des modèles de réseaux de neurones exploitant les signaux Unité de Mesures Magnétiques et Inertiels de téléphones pour faire apprendre à leurs modèles à intégrer ces données et produire les vecteurs de déplacements orientés ou uniquement le cap du piéton [Lin et al., 2020]. [Brossard et al., 2020b] propose de dé-bruiter les données IMU de véhicule via un modèle neuronal. [Brossard et al., 2020a] propose d'utiliser un modèle d'apprentissage pour apprendre à estimer la position du véhicule à partir de l'IMU seulement. Les approches par modèles d'apprentissage ont plusieurs inconvénients.

Premièrement, tout apprentissage dépend de la disponibilité de données représentatives du problème à traiter pour effectuer un entraînement. Ces données d'entraînement doivent être étiquetées pour un entraînement supervisé et d'une quantité importante afin d'éviter le phénomène de sur-apprentissage. Par exemple, le jeu de données associé à RoNIN de [Yan et al., 2019] est composé de 40 heures de séquence d'IMMU. De plus dans le cas de la localisation du piéton, il existe des caractéristiques liées au sujet qui marche quand la collecte de données est faite. Donc selon [Yan et al., 2019], pour que le système soit indifférent à l'utilisateur, il faut une variété en âge, taille, sexe et/ou poids des sujets pour que les données collectées englobent le plus de cas possibles et que la généralisation du modèle appris soit robuste. Ce qui implique un coût de construction de base de données plus élevé.

Deuxièmement, en ce qui concerne l'étiquetage de ces données, il est nécessaire d'avoir une vérité terrain qui indique la véritable position/cap du piéton. Cette vérité terrain n'est pas toujours facile à mettre en œuvre de manière à ce qu'elle soit assez fidèle. En examinant les vérités terrains proposées pour servir d'étiquette pour les données pour l'entraînement, certaines trajectoires montrent des mouvements estimés qui sont impossibles à réaliser tels que le fait de traverser des murs. Pour cette raison, ces vérités terrains sont considérées comme des pseudo-vérités terrains car elles sont des estimations plus précises du mouvement que celles du modèle entraîné. Mais, ce n'est pas pour cela qu'elles sont exactes et peuvent donc fournir des exemples erronés au système entraîné. Nous discutons des méthodes pour établir cette vérité terrain dans le chapitre 4 section 4.3.

Troisièmement, ces modèles d'apprentissage ne peuvent pas incorporer de règle sémantique. Donc, l'inclusion d'informations externes au système proprioceptifs tel que les contraintes de déplacement imposées par carte est impossible. Ce qui néglige une spécificité de la localisation en intérieur car elle se passe dans un bâtiment. L'inclusion de cartes dans le processus de localisation est une de nos contributions et est présentée en section 2.5.1.4 et en section 5.3. Cette inclusion rejoint nos objectifs [C & E].

[Luo et al., 2020b] propose d'apprendre à reconnaître le mode de déplacement pour indiquer à la PDR quelle correction utiliser sur l'angle. Le tableau 2.1 présente les résultats de leur modèle comparé à d'autres méthodes de classification du mode de mouvement .

La classification des modes de mouvement et placement de l'IMU est donc une étape préliminaire à la PDR. Une fois que le mode est déterminé, la système de localisation effectue un traitement dédié à ce mode là. Ce mécanisme de contrôle amont est connu sous le nom de

Méthode	Précision (%)	Méthode	Précision (%)
[Luo et al., 2020b]	99.72	SVM [Cortes and Vapnik, 1995]	36.25
NB [Rish et al., 2001]	37.13	KNN [Keller et al., 1985]	89.30
GBDT [Ke et al., 2017]	59.31	RF [Liaw et al., 2002]	94.10

TABLE 2.1 – Résultat de la précision de classification du mode de déplacement par [Luo et al., 2020b]

"*switching*". Dans notre cas, le mode étudié est le cas où le piéton est un marcheur qui avance. Cette thèse se concentre sur le traitement de ce seul mode.

2.2.1 Détection de pas

Un podomètre est un dispositif de détection et comptage de pas via des données inertielles. La marche étant un phénomène cyclique, le signal d'accélération est donc périodique ou pseudo-périodique [Abadleh et al., 2017]. [Orendurff et al., 2008] indique que les pas sont des séquences d'événement discrets dans une fenêtre de temps. La définition d'un pas selon [La Rousse, 2023] est : "Mouvement par lequel on porte un pied à une certaine distance de l'autre pour marcher ou courir". [Bassett et al., 2017] déduit donc que piétinement sur place ne répond pas à cette définition et doit donc être traité séparément.

Ces aspects de périodicité et de discrétisation sont retrouvés dans les signaux collectés sur l'IMU portée par le piéton où la norme de l'accélération alterne entre des pics et des vallées. Ce qui permet d'identifier le moment où un pas se produit. On s'intéresse au cas où l'IMU est maintenue au niveau du torse et non pas au cas où elle est fixée au pied (dite *Strap Down*) [Deng et al., 2017].

Une technique relativement simple et performante pour détecter le moment où un piéton effectue un pas est appelée : détection pics-vallées [Oudre et al., 2018]. Les auteurs [Ying et al., 2007] avaient adopté cette technique depuis les travaux de [Pan and Tompkins, 1985] pour la détection des pics de données électroencéphalographiques.

Cette approche est instanciée de manière similaire dans [Abadleh et al., 2017, Khedr and El-Sheimy, 2017, Ghaoui et al., 2022, Wen et al., 2023]. Les auteurs [Brajdic and Harle, 2013] reportent un taux d'erreur inférieur à 3 % pour le comptage des pas avec une étude menée sur 27 personnes, 130 marches et 6 manières de placer le téléphone. Enfin, il est aussi possible d'effectuer cette détection de pas avec le gyroscope placé dans la poche [Jayalath et al., 2013] pour calculer l'écart entre les deux jambes qui correspond à un pas. Cette technique est présentée en détail dans la section 4.1.2.

2.2.2 Estimation de la longueur du pas

Les modèles d'estimation de la longueur du pas se basent sur des concepts simples, tels que la relation entre la longueur du pas, la longueur de la jambe et l'angle d'écart de la jambe sur le plan sagittal pendant chaque pas. Néanmoins, la plupart de ces méthodes reposent sur des patterns moins évidents, qui ont été observés grâce à l'étude approfondie de la marche humaine depuis de nombreuses années, notamment dans le domaine de la biomécanique [Díez et al., 2018]. Ces modèles représentent le piéton par son centre de masse (COM) qui est confondu avec le point de pivot de la jambe sur le plan horizontale i.e. seules leurs hauteurs

sont différentes. Leur but reste le même, à partir d'un signal IMU et d'un événement de pas : fournir la longueur du pas effectué.

Les modèles de marche pour l'estimation de la longueur de pas sont présentés en détail dans la section 4.1.3.

2.2.3 Estimation du cap de pas

Les données gyroscopiques de l'IMU acquises représentent la vitesse angulaire du téléphone sur chaque axe à chaque instant d'échantillonnage. À cause du bruit présent dans ces acquisitions, intégrer simplement ces données pour avoir les angles d'Euler d'orientation du téléphone peut ne pas être valide sur des séquences assez longues. Pour corriger cette dérive, plusieurs recherches ont été menées sur l'estimation des angles à partir de données accélérométriques et gyroscopiques seulement. Corriger cette dérive est notre objectif [B].

[Yun et al., 2008] propose un algorithme simplifié qui estime l'orientation d'un appareil à l'aide de mesures provenant de capteurs de gravité terrestre et de champ magnétique. La méthode utilise des quaternions¹ pour représenter l'orientation de l'appareil et développe un algorithme efficace qui met à jour le quaternion en fonction des vecteurs de gravité et de champ magnétique mesurés. L'algorithme proposé est efficace en termes de calcul et peut être implémenté sur des appareils à faible puissance. L'article compare les performances de l'algorithme proposé avec d'autres méthodes existantes et montre son efficacité en termes de précision et de robustesse. L'algorithme est une solution simple et efficace pour l'estimation de l'orientation dans les dispositifs à faible coût et peut être appliqué dans divers domaines tels que la robotique, la navigation et la réalité virtuelle.

[Euston et al., 2008] utilise deux filtres : le premier est passe-bas sur les données d'accélération ; le second est passe-haut pour les données gyroscopiques. L'hypothèse émise est que le mouvement de rotation du piéton est associé à des transitions graduelles (relativement lente) dans le signal d'accélération, alors qu'il est associé à des variations abruptes (relativement rapides) dans le signal de vitesse angulaire.

[Mahony et al., 2008] propose un algorithme utilisé pour la fusion de capteurs, notamment pour combiner des mesures d'un accéléromètre et d'un gyroscope pour estimer l'orientation d'un appareil dans un espace tridimensionnel. Le filtre de *Mahony* est un filtre complémentaire qui utilise une technique d'optimisation non linéaire pour corriger les erreurs de dérive inhérentes aux mesures du gyroscope. Le résultat est une estimation plus précise de l'orientation de l'appareil que celle qui peut être obtenue en utilisant l'un ou l'autre des capteurs seuls.

[Madgwick et al., 2011] présente un algorithme utilisé pour la fusion de capteurs. Il combine les mesures d'un accéléromètre et d'un gyroscope pour estimer l'orientation d'un appareil dans un espace tridimensionnel. Le filtre de *Madgwick* est un filtre basé sur la descente de gradient qui utilise une représentation quaternionique de l'orientation, et il peut également prendre en compte les mesures de champ magnétique d'un magnétomètre, ce qui en fait un filtre plus avancé que le filtre de *Mahony*. Il a été démontré [Michel et al., 2018] que le filtre *Madgwick* fournit des estimations d'orientation les plus précises et stables dans une grande variété d'applications en particulier, la marche piétonne.

1. Les quaternions sont des nombres complexes à une composante réelle et trois composantes imaginaires définis par *Hamilton* [Rosenfeld, 2012]. Ils permettent de représenter des angles d'*Euler* et des rotations dans cet espace complexe s'affranchissant du blocage de cardan, connu aussi sous le nom de verrou de *Gimbal* [Hoag, 1963]. Leur étude, calcul et représentation sont en dehors de la portée de ce manuscrit.

[Sabatini, 2011] présente une approche basée sur un filtre de Kalman pour déterminer l'orientation d'un dispositif à l'aide de capteurs inertiels et magnétiques. L'article traite de l'observabilité du système et identifie le nombre minimum de capteurs requis pour atteindre l'observabilité. Les auteurs proposent un filtre de Kalman qui fusionne les mesures des accéléromètres, des gyroscopes et des magnétomètres pour estimer l'orientation de l'appareil. Les performances du filtre sont évaluées par simulation et comparées à d'autres méthodes existantes. L'article conclut que le filtre de Kalman proposé fournit des estimations précises et fiables de l'orientation de l'appareil, en particulier en présence d'erreurs de mesure et de perturbations externes. Les auteurs suggèrent que l'approche proposée peut être utile dans des applications telles que la réalité virtuelle, la robotique et l'interaction homme-machine.

[Valenti et al., 2015] propose un filtre d'orientation basé sur les quaternions à utiliser avec les IMUs et les MARGs. Le filtre est basé sur un algorithme de descente de gradient qui estime l'orientation de l'appareil dans un espace tridimensionnel à l'aide des mesures des capteurs. Les auteurs montrent que le filtre proposé fournit des estimations précises et stables de l'orientation du dispositif, même en présence d'erreurs de mesure et de perturbations externes. L'article conclut que le filtre proposé est une solution prometteuse pour améliorer les performances des algorithmes d'estimation d'orientation pour les IMUs et les MARGs.

[Wu et al., 2018] propose un algorithme rapide et précis pour déterminer l'orientation d'un appareil à l'aide de capteurs d'accéléromètre et de magnétomètre de téléphone. Les auteurs présentent une nouvelle méthode d'estimation de l'orientation de l'appareil basée sur les relations géométriques entre le champ magnétique terrestre et la force gravitationnelle. L'algorithme proposé est efficace sur le plan informatique et peut s'exécuter en temps réel sur des appareils à faible consommation tels que les portables. Les auteurs comparent les performances de l'algorithme proposé avec d'autres méthodes existantes et démontrent sa supériorité en termes de précision, de rapidité et d'efficacité énergétique. L'article conclut que l'algorithme proposé est une solution prometteuse pour la détermination de l'attitude dans les appareils grand public et peut être appliqué dans divers domaines tels que la réalité virtuelle, les jeux et la robotique mobile.

Une comparaison détaillée est présentée par [Michel et al., 2017] où les auteurs ont implanté ces algorithmes sur des téléphones et les ont testés dans une salle de capture de mouvement (*motion capture*). Ils montrent que parmi les 34 implantations comparées, le filtre de [Madgwick et al., 2011] sans magnétomètre offre un compromis correct entre précision : 4.1ř° en moyenne pour le mode "*texting*" et temps d'exécution : calcule $\approx 16\ 000$ quaternions par seconde sur le LG Nexus 5.

On souhaite utiliser uniquement les données d'accéléromètre et de gyroscope dû au fait qu'à l'intérieur d'un bâtiment, les perturbations magnétiques faussent les lectures du magnétomètre [Hu et al., 2020]. Toutes les méthodes d'estimation de cap à partir de données proprioceptives présentent un biais résiduel qui doit être compensé. Ces méthodes ont atteint des performances similaires. Nous nous intéressons à l'usage d'une de ces méthodes en tant que brique élémentaire pour suivre l'évolution du cap pour ensuite procéder à la correction de ce cap pour satisfaire l'objectif [B].

2.3 Localisation par infrastructure réseau

Comme présenté dans la figure 2.1, la localisation en intérieur peut aussi être réalisée en utilisant une infrastructure réseau soit déjà présente dans le bâtiment soit qui doit être

installée. Certaines recherches utilisent le WiFi [Roy and Chowdhury, 2022], l'identification par radio-fréquence (RFID) [Motroni et al., 2018], les systèmes à antenne Ultra Large Bande (UWB) [Caso et al., 2018, Awarkeh et al., 2018], Communication par Lumière Visible (VLC) [Dehghan Firoozabadi et al., 2019] et le Bluetooth Bas Énergie (BLE) [Faragher and Harle, 2015] pour effectuer cette localisation. Ces technologies permettent un suivi précis et en temps réel des personnes à l'intérieur des bâtiments.

Le WiFi est devenu la technologie réseau la plus populaire pour la localisation intérieure en raison de son faible coût et de sa grande disponibilité. La puissance du signal et le rapport signal/bruit sont utilisés pour estimer la position de l'utilisateur. De plus, il peut être utilisé conjointement avec d'autres capteurs pour fournir des informations de localisation plus précises [Roy and Chowdhury, 2022, Caso et al., 2018, Huang et al., 2020, Hu et al., 2017].

La RFID est une technologie qui utilise des étiquettes intégrées avec un identifiant unique pour suivre les déplacements des personnes. Ces étiquettes sont lues par un lecteur RFID et les données sont utilisées pour suivre l'emplacement de la personne [Hayward et al., 2021, Motroni et al., 2018]. Cette technologie est souvent utilisée en combinaison avec d'autres capteurs tels que des caméras et des points d'accès WiFi pour améliorer la précision du système.

L'UWB est une technologie plus récente qui gagne en popularité pour la localisation en intérieur. Elle utilise des ondes radio pour mesurer le temps de vol entre les émetteurs et les récepteurs. Cette technologie est plus précise que les autres méthodes en raison de sa capacité à mesurer des distances avec une grande précision [Caso et al., 2018, Tian et al., 2019].

Le BLE est une technologie qui devient de plus en plus populaire pour la localisation en intérieur. Il utilise des signaux Bluetooth pour détecter la présence d'objets ou de personnes, puis trianguler leur position [Faragher and Harle, 2015, Cheraghi et al., 2019]. Il est relativement peu coûteux et peut être utilisé en combinaison avec d'autres capteurs tels que des caméras et le WiFi pour augmenter la précision.

D'un point de vue déploiement, il existe un coût initial et un coût d'entretien des instruments et sous systèmes nécessaires pour le fonctionnement de l'infrastructure réseau. La localisation étant un service, il y a des exigences de continuité et de disponibilité auquel le système doit répondre et donc la consommation énergétique est aussi à prendre en compte. Et cette dernière est accompagnée de la possibilité de mise à l'échelle (*scalability*). En effet, couvrir un espace de 10 m×10 m n'est pas la même tâche que couvrir un espace 30 m×60 m.

Pour les technologies de communication, le tableau 2.2 tiré de [Obeidat et al., 2021] présente pour chaque technologie la précision et le coût correspondants. La précision moyenne est donnée en mètre. Cette précision représente un ordre de grandeur atteignable par une entrée du tableau. Le coût mêle celui du déploiement, entretien et mise à l'échelle en fonction du nombre de Point d'Accès (AP).

On peut noter que les systèmes modernes de localisation sans exception exploitent plusieurs de ces modalités et procèdent à leur fusion pour améliorer leur performances dans l'un des aspects précédemment mentionnés [Zhu et al., 2020a]. À titre de comparaison, les solutions utilisant un GNSS sont incluses sans compter le coût du satellite car les constellations sont déjà opérationnelles.

Même si on s'abstrayait du coût matériel de l'équipement réseau, le coût énergétique de ces systèmes localisations est en rapide augmentation. Déjà en 2009, aux USA, plus de 6 milliards de dollars étaient dépensés pour environ 74 TWh afin d'alimenter les structures WiFi [Jardosh et al., 2009]. Depuis, cette consommation n'a fait qu'augmenter et se mondialiser. Les prévisions de la consommation énergétique mondiale de ces dispositifs liés aux utilisateurs estiment

Technologie/Signal	Précision [m]	Coût
Sonore	5 - 10	bas
Ultra-sonore	0.01-1	moyen-haut
Proximité Infrarouge	1-2	moyen
Lumière Visible	0.1	moyen
Wi-Fi	1-5	bas-moyen
ZigBee	3-5	bas
Bluetooth	2-5	bas
UWB	0.01-1	cher-très cher
RFID	1-5	bas
Satellite	3-5	moyen en puissance seulement

TABLE 2.2 – Tableau récapitulatif des technologies de localisation dépendante de l’espace en intérieur adapté de [Obeidat et al., 2021].

que la consommation annuelle d’énergie passera d’environ 180 TWh en 2011 à 1400 TWh en 2025 (Figure 3 de [Lorincz et al., 2019]), ce qui représente une augmentation de 7.7 fois sur la période d’une décennie et demie. Un constat plus drastique est établi par [Das and Mao, 2020] où la demande d’énergie de l’industrie des communications devrait passer de 200-300 TWh d’électricité par an en 2017 à 1200 voire 3000 TWh d’ici 2025.

Ces deux aspects de communication avec l’infrastructure réseau du bâtiment et la consommation énergétique sont à l’encontre de nos objectifs [G] du fait que le système ne peut plus être considéré comme encapsulé et qu’il demande une consommation électrique plus élevée dû à l’usage du réseau.

2.4 Localisation par vision par ordinateur

La vision par ordinateur consiste en la troisième branche de la figure 2.1 où une ou plusieurs caméras fournissent leurs images à un algorithme qui produit une information extéroceptive. Dans le cas de la localisation piétonne, les caméras peuvent être statiques de façon à avoir le piéton dans leur champ de vision, ou mobiles i.e. portées par le piéton de façon à déduire le déplacement à partir du changement des images. Dans les deux cas, elles doivent communiquer à un piéton sa position et son orientation. Elles fournissent donc la pose [Antigny, 2018]. Ou bien, elles sont utilisées pour corriger l’estimation de la trajectoire.

Nous abordons avec plus de détail la vision par ordinateur appliquée à la localisation dans la deuxième partie, section 6.2 de ce manuscrit.

2.5 Fusion de données pour la localisation

2.5.1 Filtrage sur modèle d’état

Dans la section 2.1, nous avons indiqué que les systèmes modernes de localisation hybrident au moins deux modalités afin de localiser le piéton. Il faut donc un formalisme de fusion de données dans lequel les différentes informations peuvent être combinées. Dans cette partie,

l'un des formalismes les plus utilisés pour la fusion de données pour la localisation, les filtres récursifs, est présenté. Ensuite, l'usage de ces filtres dans des travaux récents est cité.

2.5.1.1 Filtre de Kalman (KF)

Le filtre de Kalman (KF) est un algorithme de filtrage optimal et récursif pour estimer l'état d'un système dynamique à partir de mesures bruitées. Il est basé sur la théorie des probabilités et utilise des modèles linéaires et gaussiens pour représenter les modèles dynamiques du système et les erreurs de mesure. Le filtre de Kalman a été initialement proposé par [Kalman et al., 1960] et depuis lors, il est devenu l'un des outils les plus utilisés pour la fusion de données et la navigation dans divers domaines tels que l'ingénierie, la robotique, l'aérospatial et la finance.

Le KF est capable d'estimer l'état présent du système en utilisant les mesures passées et les modèles dynamiques du système. Il utilise la théorie des probabilités pour modéliser la distribution de probabilité de l'état actuel et pour la mise à jour de l'estimation de l'état en temps réel. Pour l'estimation d'état non-linéaire et la représentation de l'état sous forme de processus gaussien, le KF est également utilisé en conjonction avec d'autres algorithmes. Deux variantes en résultent : le filtre de Kalman étendu (EKF) et le filtre de Kalman non-parfumé (UKF).

2.5.1.2 Filtre de Kalman étendu (EKF)

EKF a été proposé pour la première fois par [Gelb et al., 1974] et a été développé ultérieurement par plusieurs chercheurs. Le principe de l'EKF est d'approximer les modèles dynamiques non linéaires et les modèles d'observation non linéaires, toujours dans le cas de processus gaussiens, par des modèles linéaires en utilisant des approximations de Taylor. L'EKF est efficace pour les systèmes qui ont des modèles dynamiques non linéaires modérées et un bruit de mesure gaussien. Cependant, l'EKF peut être sensible à l'erreur de modélisation, en particulier lorsque le système est fortement non linéaire.

2.5.1.3 Filtre de Kalman non-parfumé (UKF)

UKF est une filtre d'estimation d'état non linéaire qui est basé sur l'échantillonnage de la distribution de probabilité de l'état. L'UKF a été proposé pour la première fois par [Julier and Uhlmann, 1997]. Le principe de l'UKF est d'approximer la distribution de l'état à l'aide de quelques échantillons représentatifs appelés points de sigma. L'UKF est plus efficace que l'EKF pour les systèmes modèles dynamiques fortement non linéaires. Ce filtre ainsi que les deux précédents supposent que la modélisation de l'état se fait par des processus gaussiens. Certaines informations extéroceptives comme celles tirées d'une carte sont des contraintes géométriques et ne peuvent être modélisées simplement sous la forme de processus gaussiens.

2.5.1.4 Filtre à particules (PF)

Le PF également connu sous le nom de filtre de Monte-Carlo, est une méthode d'estimation d'état non linéaire basée sur la simulation stochastique de l'évolution de la distribution de probabilité de l'état. Il a été proposé pour la première fois par [Gordon et al., 1993]. Le principe du PF est de représenter la distribution de probabilité de l'état par un ensemble de particules, chacune ayant un poids associé représentant sa vraisemblance. Il ne s'agit plus d'un processus

gaussien. le PF abandonne l’hypothèse de gaussianité. Les particules sont propagées dans le temps en utilisant les modèles dynamiques du système prenant compte d’un bruit de processus d’innovation. Les poids des particules sont mis à jour à l’aide d’information extéroceptive et sont normalisés pour obtenir une distribution de probabilité.

Le PF est efficace pour les systèmes à base de modèles dynamiques hautement non linéaires et pour les situations où la distribution de probabilité de l’état est multimodale. Il permet également le suivi multi-hypothèses. Il est également robuste aux erreurs de modélisation. Cependant, il peut être coûteux en termes de calculs, en particulier pour des systèmes de grande dimension.

2.5.1.5 Récapitulatif sur les filtres

En résumé, les filtres de Kalman et leurs extensions ont été largement utilisés pour l’estimation d’état dans une variété de domaines. Dans le cas de processus gaussiens non linéaires, l’EKF et l’UKF sont utilisables. Cependant, ces filtres ne sont pas utilisables pour les processus non gaussiens. Le PF a été développé pour remédier à cette limitation en utilisant un processus discret à base de particules. Chaque méthode a ses avantages et ses inconvénients en termes de précision, d’efficacité de calcul et de sensibilité à l’erreur de modélisation. Le choix de la méthode dépend des caractéristiques du système et des exigences de l’application.

2.5.2 Usage dans la localisation

Dans cette sous-section, on s’intéresse à savoir comment les filtres précédemment mentionnés sont utilisés dans la recherche. On s’intéresse particulièrement à la fusion multimodale de signaux inertiels avec des informations extéroceptives. Ces filtres sont aussi déployés en avant d’autres algorithmes. Par exemple, [Hu et al., 2020] utilisent un arbre de décision pour juger si le KF peut exploiter la donnée du magnétométrie si elle est gaussienne sur une fenêtre de temps. [Fan et al., 2019] ont proposé un filtre complémentaire pour supprimer les erreurs cumulées d’accéléromètre et de magnétomètre suivi d’un KF robuste et adaptatif qui annule l’interférence des valeurs aberrantes car la valeur du cap reçue depuis le filtre complémentaire est sensée être optimale.

Les auteurs [Hou and Bergmann, 2020a] ont proposé de poser l’IMU sur des lunettes et dans une gouttière buccale. Ils utilisent un EKF afin de localiser le porteur et de comparer les deux dispositions. Un EKF est utilisé par [Deng et al., 2015] pour estimer l’orientation d’un téléphone posé dans la poche. Ils projettent les signaux d’accélération sur le plan horizontal rectifié via la matrice de rotation obtenu par l’EKF. Ces signaux projetés sont décomposés suivant une analyse en composantes principales. Enfin, ils recalent le cap à chaque fois que l’utilisateur produit un virage. Les auteurs [Tong et al., 2020] utilisent un double UKF pour faire une mise à jour en deux temps : la première exploite les données d’accéléromètre et évite la correction cap ; la seconde exploite les données du magnétomètre et évite la correction des roulis et tangage.

Pour le PF, les auteurs [Gu et al., 2015] l’ont déployé pour fusionner l’information de déplacement venant d’une PDR à la direction dominante du cap dans le bâtiment car ils considèrent que la plupart des bâtiments qui sont des parallélépipèdes ont des directions dominantes selon l’orthogonalité des couloirs. Cela peut donc servir à contraindre le cap du piéton lors du mouvement.

[Liang et al., 2018] fusionnent dans le PF les signaux d'accélération, gyroscope, magnéto-mètre, WiFi et lumineux avec 3 cartes de signaux : WiFi, lumineux et magnétiques. Ces cartes ont été construites en pré-calculant les trajectoires optimisées possibles durant une phase hors ligne. Ces trajectoires optimisées sont calculées via un GraphSLAM [Huang et al., 2011].

En termes de robustesse, [Blok et al., 2019] ont effectué la validation d'une localisation via KF et une autre via PF dans des circonstances comparables pour permettre une navigation robotique précise et robuste dans les vergers. Ils proposent une nouvelle méthode pour évaluer la robustesse de navigation des robots dans des conditions réelles de verger. Ils ont testé des modèles d'arbres pour évaluer comment les performances du robot étaient dégradées lorsque certains arbres manquaient. Le PF a surpassé le KF en termes de précision et de robustesse de la navigation avec une erreur de positionnement moyenne 4.35 cm contre 6.93 cm pour KF. En termes d'erreur angulaire, les auteurs reportent que les deux algorithmes ont eu les mêmes performances.

Les auteurs de [Hafner et al., 2013] ont mené une étude comparative de plusieurs filtres bayésiens pour la fusion de la carte, de l'IMMU et des émetteurs radio dans le contexte d'une modélisation par un espace de navigation borné par une carte. Seul le PF fusionne la carte du bâtiment aux autres modalités. Le PF permet dans ce cas de supprimer les particules produisant un mouvement interdit ou impossible (e.g. traverser un mur) et de renforcer celles qui ne les franchissent pas en les dupliquant. Le KF fonctionne uniquement dans un environnement gaussien et en cas de système et équations de mesures linéaires, les performances pour une application en intérieur sont limitées, puisque l'intégration des plans de construction entraîne une modification de la distribution gaussienne [Hafner et al., 2013]. C'est pour cela que [Taia Alaoui, 2018] dit que « au bilan, le KF ne se prête pas à l'utilisation d'un espace borné dans une carte [contrairement au PF]. » Le PF est donc bien adapté à la fusion de contraintes géométriques provenant d'une carte de l'étage. Nous expliquerons l'interaction entre le PF et la carte plus précisément dans la section 5.3.

Une approche mêlant le KF et le PF est proposée par [Yu et al., 2017]. Le KF produit la fusion des données inertielles suivant l'approche "*strapdown*", où l'IMU est attaché au pied, et des mises à jour à vitesse nulle. Il fournit la longueur et le cap du pas à un PDR qui les remets ensuite au PF. Le PF incorpore les contraintes de la carte. [Yu et al., 2017] nomme cette étape "Assistance par carte" (*Map Aiding*) pour définir quelle particule est valide ou non. Ensuite, l'étape de correspondance sur la carte est appliquée (*Map Matching*). Il nomme ce double filtre : filtre auxiliaire.

On note donc que les contraintes spatiales et géométriques sont intégrées au PF pour lui permettre d'exploiter la carte du bâtiment durant la localisation [Beauregard et al., 2008]. Dans [Ning and Chen, 2019], deux cartes sont utilisées : la première contraint le mouvement des particules ; l'autre est un relevé d'intensité magnétique dressé au-préalable afin que le système puisse réaliser une recherche de correspondance durant la localisation. Ce même concept est appliqué par [Huang et al., 2020], en remplaçant l'intensité du magnétomètre par l'intensité du signal WiFi. Dans [Zhao et al., 2019], le PF est toujours contraint par les murs de la carte, mais cette fois, le baromètre est inclus pour déterminer à quel étage est le piéton pour savoir quelle carte il faut utiliser. L'idée est reprise par [Lu et al., 2019] à la différence que leur PF ne prend en compte les lectures du baromètre que dans les zones de la carte permettant des transitions entre étages. Ils appliquent aussi un "retour en arrière" (*Backtracking*) si la position estimée par le PF est impossible de manière similaire à [Klepal et al., 2008]. [Zhou et al., 2017] utilisent la carte pour contraindre la trajectoire estimée par le PF sur un graphe de l'espace

intérieur qu'ils ont construit.

L'un des reproches que le PF reçoit est qu'il peut avoir un coût calculatoire élevé si le nombre de variables d'état à modéliser est élevé. Dans les publications [Lu et al., 2019, Zhao et al., 2019, Liang et al., 2018, Yu et al., 2017, Ning and Chen, 2019], les auteurs différaient tous par le nombre de particules utilisées pour leur PF, respectivement 256, 500, 1000, 1000, 1000 particules. Cela indique que, pour ce qui semble être le même problème, soit toutes les approches ne sont pas très efficaces avec leur utilisation de particules, soit le problème traité a une couche cachée de complexité. Certains statisticiens proposent des techniques pour avoir un nombre de particules adaptatif [Elvira et al., 2017, Ahwiadi and Wang, 2020].

2.6 Synthèse

De tous les points précédemment présentés, on retient que la PDR est une technique simple reposant sur les données inertielles. Elle représente un premier résultat qui donne l'allure de la trajectoire entreprise par le piéton. Les limites de cette technique sont liées à une première problématique qui est celle de l'estimation de la longueur du pas où plusieurs modèles ont été présentés en recherche : biomécaniques, paramétriques et empiriques. Ces limites sont la raison de l'existence des objectifs [A & B].

Nous avons présenté un résumé des technologies de localisation en intérieur qui reposent sur l'infrastructure réseau d'un espace intérieur pour localiser le piéton. Ces technologies se reposent sur une ou plusieurs modalités et, en appliquant une technique, déterminent la position de l'appareil dont le piéton est équipé. En terme de précision, on retrouve déjà une erreur de localisation inférieure à celle des systèmes GNSS i.e <5 m. Certaines d'entre elles, notamment l'UWB et l'ultrasonique, présentent une précision centimétrique. Ce qui serait déjà satisfaisant pour de nombreuses applications.

En termes de déploiement, les AP WiFi couvrent la quasi-totalité des bâtiments de nos jours. Donc, une localisation par cette modalité est attractive du fait que le coût de déploiement est pratiquement nul. Le BLE est lui aussi un bon prétendant car il est le plus simple à déployer, ses modules ont un prix qui défie la concurrence pour la précision demandée. Ceci rend la redondance des modules Bluetooth moins chers à réaliser et peut donc augmenter la robustesse de la localisation. De plus ces deux technologies sont déjà très bien intégrées dans tous les téléphones de toutes les gammes.

Il semble à première vue que les solutions existantes sans être fusionnées sont déjà satisfaisantes. Toutefois, elles peuvent nécessiter un sondage de l'espace en intérieur pour construire la base de données des signaux de référence et leur position de référence si elles utilisent le Received Signal Strength (RSS). Cette opération est l'une des plus coûteuse en temps et en argent et doit être faite bien à l'avance, i.e. elle est inutilisable dans le cas d'interventions dans des lieux inconnus.

À l'heure où la sobriété énergétique est l'un des sujets les plus pertinents, la généralisation de systèmes de localisation se basant sur des infrastructures réseaux pose un problème. En effet, ces systèmes vont fortement augmenter cette consommation. Donc, il y a un effort important en recherche et en industrie à concevoir des systèmes de localisation qui minimisent cette consommation énergétique voire à l'annuler en s'abstenant d'utiliser les réseaux de communication.

D'un point de vue applicatif, les systèmes reposant sur le réseau d'un bâtiment sont simplement incompatibles avec les applications qui traitent de l'urgence. Ces applications demandent

à ce que le piéton soit localisé instantanément, sans préparation d'infrastructure au préalable dans le bâtiment, par exemple, durant une coupure électrique ou durant un incendie. Elle demande aussi à pouvoir couvrir la totalité de l'espace en intérieur tout le temps. Les systèmes reposant sur des réseaux ne peuvent simplement pas garantir ces deux exigences car leur dépendance au réseau du bâtiment est intrinsèque à leur fonctionnement.

Enfin, deux des points les plus relevés durant cette décennie sont : vulnérabilité et confidentialité. Tout système communicant s'expose à un risque de cyber-attaque. Et donc il expose son utilisateur à une fuite de ses données de déplacement, à une attaque pour le fauter et/ou le mener à un danger. En terme de confidentialité, celle-ci est devenue l'axe de beaucoup d'entités économiques ou politiques qui promettent de garantir le non-usage des données de leurs utilisateurs sans leur accord. Étant donné que les systèmes présentés auparavant communiquent avec le réseau du bâtiment, tout utilisateur s'interroge sur le traitement des ses données.

C'est pour toutes ces raisons qu'une partie de la recherche en localisation en intérieur s'est portée sur des solutions totalement embarquées, encapsulées, autonomes et non-dépendantes du réseau de l'espace intérieur en alliant les IMUs et la caméra embarquées pour répondre à l'objectif [G].

En termes de fusion de données, l'usage de filtres permettant de fusionner les signaux inertiels avec une information externe a été présenté. Le filtre à particules semble être l'outil adapté à la localisation basée sur une PDR qui est corrigée par l'inclusion d'une carte du bâtiment. Le filtre à particules (PF) permet d'inclure cette carte dans le processus de fusion car il est adapté aux traitement des problèmes non gaussiens.

La gestion multi-hypothèses que permet le PF est un aspect que l'on considère important. Chacune de ses particules représente une hypothèse et dans le cas de la marche piétonne, une trajectoire possible. Il permet donc de représenter l'espace des possibles à l'instant d'un pas et de suivre son évolution dans le temps. Ceci est fait en appliquant différents poids à chaque particule en fonction de leur vraisemblance. Ceci rejoint notre troisième objectif : modéliser la vraisemblance du mouvement humain dans des espaces intérieur. Enfin, le PF utilise un nombre fini de particules pour modéliser la densité *a posteriori*. Il peut donc offrir un compromis précision d'estimation vs temps d'exécution en variant le nombre de particules.

2.7 Structure de la partie I

Dans le chapitre 3, nous définissons les différents systèmes de coordonnées utilisés pour notre étude. Ces systèmes de coordonnées suivent les hypothèses prises pour cette thèse qui sont présentées dans la section 3.3. Plusieurs tests de validation d'acquisition sont faits afin de garantir la cohérence des données acquises par rapport au mouvement physique entrepris.

Nous présentons dans le chapitre 4 notre usage de la méthode de *Weinberg*, du filtre de *Madgwick* ainsi que la PDR. Une série de tests de validation est effectuée dans la section 4.1. Ces tests visent à vérifier le fonctionnement correct de la détection des événements du pas, de la distance parcourue via PDR sur des trajectoires rectilignes 2D, curvilignes 2D, et 3D en intégrant la hauteur, du suivi d'orientation 3D et de l'invariance aux utilisateurs.

Dans le chapitre 5, nous procédons à une étude détaillées du PF en section 5.1. Nous expliquons les étapes nécessaires au fonctionnement de ce filtre et les modifications que nous avons apportées. C'est parmi ces étapes que nous introduisons les contraintes spatiales tirées de la carte du bâtiment. Pour éviter un cas problématique liée au calcul de la moyenne pondérée qui résulte dans une zone impossible du bâtiment, une solution est proposée en section 5.3.5.

Nous présentons dans la section 5.3 une autre de nos contributions : la régénération qui remplace le ré-échantillonnage et qui tient compte des contraintes spatiales de la carte. Cette régénération a pour rôle de rajouter les particules manquantes ayant été supprimées en utilisant les contraintes géométriques issues de la carte car elles ont effectuées un mouvement interdit.

Nous avons aussi procédé à l'étude de l'impact du nombre de particules sur les performances du système dans la section 5.5.1. Les méthodes d'évaluation proposées visent à mesurer la précision du système, l'intégrité de la position estimée et la robustesse du paramètre de *Weinberg*.

2.7.1 Contributions

Enfin, notre contribution la plus importante de [Ghaoui et al., 2022] est la modélisation de la vraisemblance du mouvement humain dans les espaces intérieurs sous la forme d'une grille construite à partir de la carte du bâtiment. Cette grille permet d'assigner un poids aux particules du PF ce qui permet de représenter la vraisemblance qu'une particule occupe une position. La grille de vraisemblance permet de resserrer les ellipsoïdes d'incertitude. Dans la finalité, cette grille de vraisemblance peut nous permettre de décrire des comportements de déplacement pour le piéton dans l'espace intérieur.

Notre deuxième contribution est la méthode d'évaluation du système proposé en termes de précision, robustesse et intégrité. Dans [Ghaoui et al., 2022], il est démontré que cette méthode d'évaluation a une vérité terrain satisfaisante en résolution spatiale et temporelle.

Les codes et base de données liés à nos expériences de cette partie ont été publiés avec [Ghaoui et al., 2022] et sont disponibles sur <https://github.com/anisghaoui/humolire>.

Chapitre 3

Acquisitions des données, systèmes de coordonnées et tests de validation

Sommaire

3.1	Acquisition de données	44
3.1.1	Alignement Spatial	44
3.1.2	Alignement Temporel	45
3.1.3	Calibration des capteurs	45
3.2	Systèmes de coordonnées - SC	47
3.2.1	Système de coordonnées global - SCG	47
3.2.2	Système de coordonnées utilisateur - SCU	47
3.2.3	Système de coordonnées appareil - SCA	48
3.3	Hypothèses prises pour cette thèse	51
3.3.1	Position et orientation de l'appareil par rapport à l'utilisateur	51
3.3.2	Verticalité du piéton	52
3.3.3	Mise à jour événementielle	52
3.4	Tests de validation	52
3.4.1	Test d'acquisition statique	52
3.4.2	Calibration et non-calibration par Android	53
3.4.3	Tests des translations unitaires	54
3.4.4	Tests des rotations unitaires	58
3.5	Récapitulatif	61
3.6	Conclusion	62

Avant de procéder à notre étude, il est primordial de définir les bases nécessaires pour faire fonctionner nos algorithmes sur des données cohérentes et les mesures utilisées pour juger de leurs performances. Dans ce chapitre, la plateforme et les outils d’acquisition des données sont spécifiés. Par la suite, trois systèmes de coordonnées sont définis : celui du piéton, celui de l’appareil qu’il porte et celui du bâtiment dans lequel il se déplace. Les métriques d’évaluation sont présentées après cela. Enfin, nous exposons les tests de validation conduits pour valider l’acquisition des données inertielles.

3.1 Acquisition de données

Afin d’acquérir des données inertielles, le choix d’une plateforme d’acquisition pour l’utilisateur est indispensable. En accord avec notre objectif [F], les téléphones portables sont omniprésents et embarquent tous une Unité de Mesures Inertielles (IMU). L’IMU est légère, n’est pas énergivore et possède une précision satisfaisante pour les applications visées par le téléphone [Ashraf et al., 2020]. Plus précisément, elle embarque : un accéléromètre, un gyroscope. Si elle inclut aussi un magnétomètre, elle devient une IMMU. Celui-ci n’est pas utilisé dans nos travaux comme expliqué dans la section 2.1.

Afin d’interfacer cette IMU, une application doit être installée sur un téléphone Android. Il en existe plusieurs¹ et elles répondent différemment à plusieurs critères en fonction du besoin de l’utilisateur. Le tableau 3.1 liste celles qui ont été retenues et testées dans le cadre de nos expériences, les capteurs acquis, l’ordre de grandeur de l’erreur de synchronisation des données et une particularité qui concerne l’application.

Nom	Capteurs	Synchronisation	Particularité
SensLogs [Michel et al., 2018]	Tout capteur disponible	à $0.5 \mu s$ près	Impossibilité de fixer les paramètres camera
MarsLogger [Huai et al., 2019]	IMU + vidéo	à $1 ms$ près	Possibilité de fixer les paramètres caméra
RoNIN DataLogger [Yan et al., 2019]	IMU + Android Game Vector	Horloge système	Contrôle limité sur l’application

TABLE 3.1 – Liste des applications d’acquisition de données retenues puis testées.

Nous choisissons d’utiliser l’application MarsLogger pour les expériences de ce manuscrit afin d’acquérir les données inertielles et caméra. L’application permet de fixer le point focal et l’exposition de la caméra avant chaque prise. La fréquence d’acquisition est fixée à 200 Hz par l’application.

3.1.1 Alignement Spatial

L’alignement spatial des capteurs est la caractéristique qui assure que les capteurs sont correctement positionnés les uns par rapport aux autres lors de l’acquisition. Elle assure aussi le fait que les données acquises soient cohérentes avec le déplacement. Dans le cas du téléphone, cet alignement est fait par le constructeur pendant la pose des composants embarqués dans l’architecture. En notant que les téléphones font une quinzaine de cm de haut, l’erreur d’alignement en translation est négligeable comparée à l’erreur de localisation du piéton. L’erreur d’alignement en orientation entre le téléphone et le capteur quand à elle est plus problématique. Il est nécessaire de procéder à un test unitaire de validation des données gyroscopiques acquises. Si, l’on considère que ces erreurs d’alignement spatiales sont négligeables, tous les

1. Voir en Annexe section A.1

repères des capteurs embarqués peuvent être confondus avec le repère égocentré du téléphone. On ne représente alors que le repère téléphone comme dans la figure 3.2.

3.1.2 Alignement Temporel

L’alignement temporel (aussi appelé synchronisation) des capteurs est la caractéristique qui assure que les capteurs font leurs acquisitions au même instant physique t . i.e. que ces capteurs ont la même origine temporelle (le même début) et qu’ils sont cadencés à la même vitesse (fréquence) d’acquisition. Cette synchronisation garantit que les données sont temporellement cohérentes. Il existe deux moyens de réaliser cette synchronisation.

3.1.2.1 Synchronisation Matérielle

Dans le cas où il est possible d’utiliser une source de cadencement unique (une horloge commune) et monotonique (oscillateur par impulsions), les données de tous les capteurs sont estampillées par le même horodatage lors d’une acquisition. La différence qui peut exister entre les estampillages des différents capteurs acquis au même instant est liée à la latence de la réponse du système de datation. Cette latence dépend de la programmation du système d’acquisition et de son fonctionnement interne. Son ordre de grandeur doit être beaucoup plus faible que la période d’échantillonnage pour un problème traité. La synchronisation matérielle garantit par construction que les données seront alignées sur la même source temporelle. Mais, son désavantage est qu’elle n’est pas toujours faisable et ne peut pas être rajoutée après la production de l’appareil.

3.1.2.2 Synchronisation Logicielle

Il est parfois impossible d’avoir accès à une source temporelle unique dans un système. Dans ce cas, on utilise plusieurs sources temporelles que l’on synchronise *a posteriori*. Cet usage relève 2 problèmes :

- Décalage à l’origine : plusieurs horloges n’ont pas nécessairement le même instant de départ. Ce décalage est simple à résoudre car il s’agit d’une translation temporelle d’une source par rapport à l’autre.
- Déviation en fréquence : due aux imperfections liées aux non-linéarités des oscillateurs, les horloges dévient en fréquence en fonction du temps. Autrement dit, le temps peut s’écouler différemment d’une horloge à une autre. Cette déviation est problématique et n’est pas triviale à corriger.

Dans notre cas, le système Android assure l’estampillage des données des capteurs par l’horloge système unifiée.

3.1.3 Calibration des capteurs

L’IMU est usinée sous la forme d’un circuit intégré qualifié de Système Micro-Électro-Mécanique (MEMS) dont les dimensions sont inférieures à $100 \mu m$ [Chollet and Liu, 2013] et qui allie les principes mécaniques et électroniques pour mesurer une grandeur physique.

Une IMU est dite 6-axes quand elle embarque 3 accéléromètres et 3 gyromètres uni-axiaux pour fournir des mesures en 3D. Ces capteurs sont disposés afin de former 2 bases orthogonales dans \mathbb{R}^3 avec les axes des capteurs alignés à ces bases. Dans les téléphones, les IMUs utilisées

sont à bas coûts. La fabrication de leurs capteurs n'étant pas parfaite, plusieurs erreurs lors des mesures peuvent apparaître.

3.1.3.1 Erreurs de mesures des capteurs

Facteur d'échelle. Pour toute valeur physique que l'on souhaite mesurer, il existe une grandeur retournée par le capteur qui n'est pas forcément exactement la même. Cette différence est nommée le facteur d'échelle qui peut être propre à chaque axe de capteur.

Couplage inter-axes. Le processus de fabrication d'une IMU vise à disposer les axes de manières orthogonales. Ainsi, en théorie, les 3 axes sont totalement découplés. Dans la pratique, cette fabrication est imparfaite et introduit des effets de couplages pour une mesure.

Biais. Il peut être défini par le décalage à l'origine de mesure du capteur. i.e. la valeur que renvoie le capteur quand il est au repos. Une majeure partie du biais est constante, donc simple à éliminer. Alors qu'une autre partie peut varier dans le temps et être affectée par différents facteurs comme la température du MEMS.

Bruits de mesure. Enfin, toute acquisition possède un bruit de mesure d'origine électrique et/ou mécanique. Ce bruit est additif et est modélisé par un bruit blanc.

3.1.3.2 Processus de calibration

La calibration est le processus visant à déterminer les différentes valeurs des erreurs modélisées afin de les compenser ou corriger. Il s'agit souvent d'une suite d'actions ou inactions protocolaires faites par l'utilisateur ou de calculs sur les toutes premières séquences de données automatiquement faites par le système avant de permettre l'acquisition.

Il se trouve que l'API d'Android² permet de produire des requêtes de données IMU déjà calibrées³ ou non-calibrées⁴ accompagnées par les biais estimés par Android. Dans leur papier [Michel et al., 2017] mentionnent que : "Les algorithmes de calibration dans iOS et Android ne sont pas publics et nous les considérons donc comme des "boîtes noires"". C'est pour cela que, dans leurs travaux [Michel et al., 2015, Michel et al., 2017, Michel et al., 2018], ils ont procédé eux mêmes à la calibration sans recourir à celle présente sur Android. Il est important de noter que dans leurs travaux, il y a un usage important du magnétomètre.

Le tableau 3.2 de [Michel et al., 2015] compare les erreurs moyennes absolues et leurs écart-types des valeurs en sorties du magnétomètre entre trois méthodes de calibration : la non-calibration, avec la calibration Android, la calibration [Renaudin et al., 2010] pour le magnéto et la calibration [Renaudin et al., 2010] + [Frosio et al., 2008] pour l'accéléromètre.

On remarque que la calibration du magnétomètre réduit très fortement l'erreur d'estimation de l'orientation du téléphone à elle seule (plus de détails⁵). Alors que la calibration de l'accéléromètre avec [Frosio et al., 2008] ne réduit que de peu les mêmes mesures. Ceci implique que dans leur étude la calibration du magnétomètre avait un impact beaucoup plus important que celle du gyroscope ou de l'accéléromètre. Les auteurs [Michel et al., 2015] expliquent cela par la possibilité que iOS/Android utilise une calibration de magnétomètre dite

2. 3. 4. Voir Annexe section A.1

5. <http://tyrex.inria.fr/mobile/benchmarks-attitude/#comparison-calibration>

	non calibré	Android	[Renaudin et al., 2010]	[Renaudin et al., 2010] + [Frosio et al., 2008]
Erreur moyenne absolue [°]	92.64	10.53	6.05	5.27
Écart-type [°]	41.13	6.41	3.58	3.50

TABLE 3.2 – Tableau comparatif de l’estimation de l’orientation d’un téléphone par [Michel et al., 2015].

"à la volée". Cette calibration est souvent satisfaisante mais il se peut qu’elle se produise au mauvais moment menant à un résultat d’estimation fortement dégradé.

Enfin, il est important de noter que les recherches [Michel et al., 2018] ont été effectuées avant 2017 i.e. avant l’avènement de ARKit par *Apple* et ARCore par *Google* mentionnés dans le blog de *Matt Miesnieks* [Matt, 2017b]. Ces deux produits demandent une calibration bien plus fine que celle des téléphones [Matt, 2017a]. On peut donc imaginer qu’en 2023, les téléphones actuels, avec l’évolution des algorithmes d’*Android/iOS*, n’ont plus besoin de calibration supplémentaire au delà du magnétomètre qui est faite à la volée. Cette hypothèse est testée dans la section 3.4.2. Le but est de valider le processus d’acquisition via l’application *MarsLogger* que nous avons sélectionnée. Ce processus devient une brique élémentaire dans notre système.

3.2 Systèmes de coordonnées - SC

3.2.1 Système de coordonnées global - SCG

Le piéton évolue dans un espace en intérieur : un bâtiment. Ce SC représente le système de coordonnées dans un bâtiment. Il est donc indispensable de ramener tout autre repère à celui-ci pour effectuer la localisation. Le piéton est représenté par un état E_g composé de deux termes : position 2D $P_g = [x_g, y_g]$ et cap de corps θ_g via l’équation (3.1).

$$E_g = \{P_g = [x_g, y_g] \quad , \quad \theta_g\} \quad (3.1)$$

3.2.2 Système de coordonnées utilisateur - SCU

Le piéton est considéré comme l’entité à localiser. Il existe un système de coordonnées qui est relatif à son corps pour tout objet équipant le piéton. Les déplacements SL et différence de cap $d\theta$ estimés par la PDR sont représentés dans ce SC. L’état E_u d’un corps dans le SCU est donné par l’équation (3.2).

$$E_u = \{P_u = [x_u, y_u] \quad , \quad \theta_u\} \quad (3.2)$$

Donc l’état définissant le piéton dans ce SC est

$$E_u^0 = \{[0, 0] \quad , \quad 0\} \quad (3.3)$$

Pour le passage des position et cap de l’utilisateur depuis SCU dans le SCG, une transformation $C_{u \rightarrow g}$ est nécessaire.

$$E_g = C_{u \rightarrow g}(E_u) \quad (3.4)$$

Cette transformée est composée d'une matrice rotation $\mathbf{R}_{u \rightarrow g}(\theta)$ autour du cap $\theta = \theta_g$ et du vecteur de translation $\mathbf{T} = [t_x, t_y]$. La rotation dans l'équation (3.5) est uniquement autour de l'axe Z d'un angle θ comme le montre la figure 3.1 Cette propriété provient de l'hypothèse présentée dans la section 3.3.

$$P_g = \mathbf{R}_{u \rightarrow g}(\theta) \cdot P_u + \mathbf{T}_{u \rightarrow g} \quad (3.5)$$

$M.V$ dénote le produit entre une matrice M et un vecteur V .

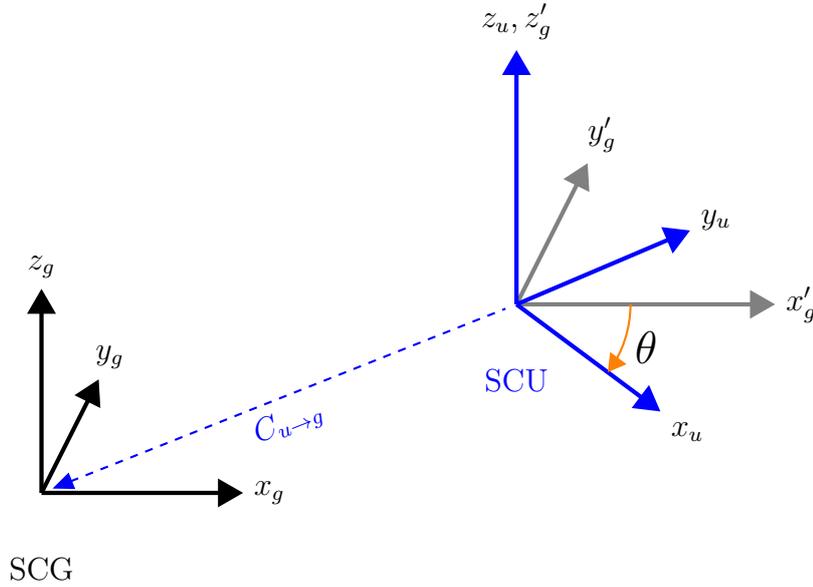


FIGURE 3.1 – Changement du SC Utilisateur à SC Global.

Depuis l'équation (3.5), on obtient :

$$E_g = \left\{ P_g = \begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} x_u \\ y_u \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \quad \theta \right\} \quad (3.6)$$

Donc, localiser un piéton dans un bâtiment revient plus formellement à trouver et suivre les valeurs de la translation $[t_x, t_y]$ et le cap θ en fonction du temps.

3.2.3 Système de coordonnées appareil - SCA

Le piéton porte un appareil équipé de capteurs. Le placement de l'appareil peut avoir un effet majeur sur les contraintes appliquées au système de localisation. Par exemple, la poitrine est une zone du corps qui ne bouge pas trop brusquement contrairement à la main. Les données brutes des capteurs sont représentées dans ce SC. Un corps dans ce SC est représenté par l'état E_a :

$$E_a = \{[x_a, y_a, z_a] \quad , \quad [\phi, \psi, \alpha]\} \quad (3.7)$$

Pour obtenir la position de l'appareil dans le SCG, une transformation de SCA à SCG, notée $C_{a \rightarrow g}$, est nécessaire. Ce changement de coordonnées peut se faire en deux temps : en premier, passer du repère appareil vers le repère utilisateur :

$$P_u = C_{a \rightarrow u}(P_a) \quad (3.8)$$

En second, appliquer la transformation de SCU à SCG comme vu précédemment dans l'équation (3.5).

Le changement de système de coordonnées de SCA à SCU est illustré à la figure 3.2 et est défini par l'équation (3.9) :

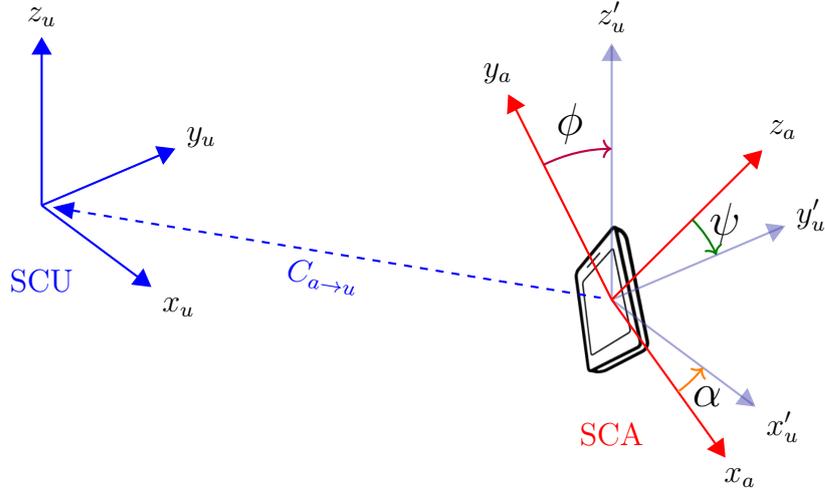


FIGURE 3.2 – Transformations de SCA à SCG.

$$E_u = \{P_u = \mathbf{R}_{a \rightarrow u} \cdot P_a + \mathbf{T}_{a \rightarrow u} \quad , \quad \theta_u = \alpha\} \quad (3.9)$$

On note que dans la figure 3.2 les axes y_a et z_a du repère téléphone correspondent respectivement à z_u et y_u dans le repère piéton. Et, selon l'hypothèse présentée en section 3.3, seul l'angle α est présent dans le SCU.

Depuis l'équation (3.9), on obtient :

$$P_u = \begin{bmatrix} x_u \\ y_u \\ z_u \end{bmatrix} = R_{a \rightarrow u} \cdot \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} + T_{a \rightarrow u} \quad (3.10)$$

Où :

1. $T_{a \rightarrow u}$ est la translation liée au placement de l'appareil sur le corps.
2. $R_{a \rightarrow u}$ est la matrice de rotation définie par le produit des 3 matrices de rotations pour chaque angle d'orientation dans l'ordre suivant :

$$R_{a \rightarrow u} = R_\alpha \cdot R_\psi \cdot R_\phi \quad (3.11)$$

$$R_{a \rightarrow u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

À partir des équations (3.5) et (3.9), on obtient la transformation finale dans l'équation (3.13).

$$E_g = \{P_g = R_{u \rightarrow g} \cdot (R_{a \rightarrow u} \cdot P_a + T_{a \rightarrow u}) + T_{u \rightarrow g} \quad , \quad \theta = \alpha\} \quad (3.13)$$

Le schéma complet est présenté par la figure 3.3.

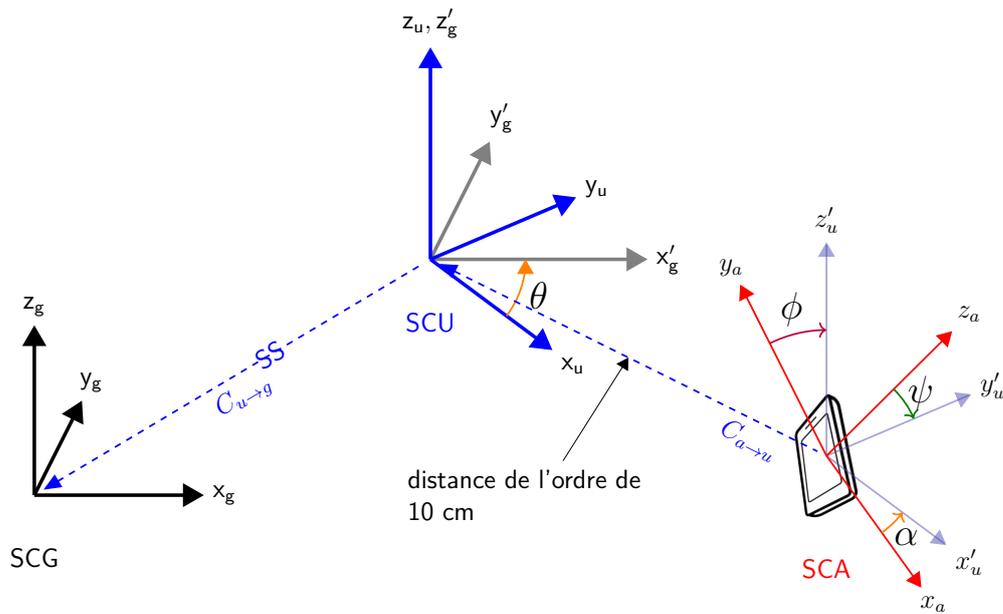


FIGURE 3.3 – Relation entre les différents Systèmes de Coordonnées.

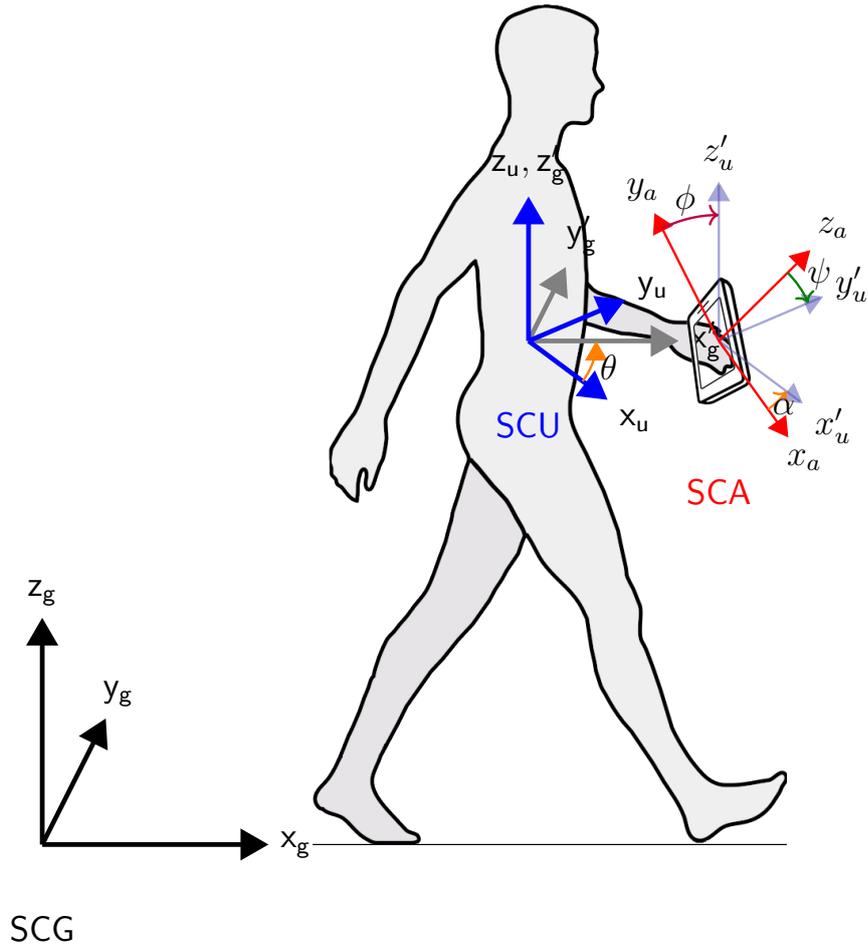


FIGURE 3.4 – Le piéton et les SCs.

3.3 Hypothèses prises pour cette thèse

Après avoir défini les différents systèmes de coordonnées, il existe des hypothèses qui peuvent être prises sans remettre en cause la cohérence des données par rapport à la physique du mouvement.

3.3.1 Position et orientation de l'appareil par rapport à l'utilisateur

Dans nos expériences, le téléphone est tenu en main au niveau du torse du piéton dans une position verrouillée à environ 10 cm de la poitrine. De cette manière l'axe du port USB dans le plan sagittal pointe vers le sol. L'axe de la caméra arrière pointe vers l'avant du piéton. Ceci implique que nous considérons les angles ψ et ϕ de la figure 3.2 constants. Dans ce cas, le cap du téléphone et de l'utilisateur sont supposés être identiques. Cette hypothèse est valide dans plusieurs contextes applicatifs dans lequel l'IMU est attachée à un agent : la navigation de personne malvoyante [Fusco et al., 2020], l'utilisation de vêtements ou équipements avec IMU et caméra intégrées [Hou and Bergmann, 2020a, Qiu et al., 2021].

3.3.2 Verticalité du piéton

Nous supposons aussi que le piéton reste à la verticale i.e. le haut de son corps ne se penche pas sur le côté. Cela implique que ses angles de roulis et de tangage sont constants tout au long du déplacement. Cette hypothèse est valide car la marche bipède demande de garder le corps en équilibre sur toutes les phases du cycle de marche vues en section 2.1. Si le piéton se penche, il sort de son équilibre, au risque de tomber. Si le piéton est toujours à verticale pendant la marche, cela implique qu'il a la même verticalité que le bâtiment dans lequel il est. C'est sur ce constat que repose cette hypothèse.

3.3.3 Mise à jour événementielle

La détection des pas est un processus de discrétisation de la marche qui est un phénomène continu. Tout le système est cadencé par les événements issus de cette détection. Et donc, il n'y a aucun calcul de changement dans son état entre deux mises à jour. L'acquisition des données est quant à elle cadencée à une fréquence plus importante.

3.4 Tests de validation

Dans le but de vérifier si les données acquises sont correctes, nous procédons à une série de tests de validation. Chacun des tests a pour but de déterminer si une procédure fonctionne en accord avec un protocole et un résultat attendu de manière toujours répétable.

Pour les expériences ci-dessous, l'acquisition des données se fait par l'application Android MarsLogger [Huai et al., 2019]. L'application permet de collecter des données IMU et caméra tout en fixant le point focal, l'exposition et l'ISO pendant la même séquence d'acquisition. Elle garantit aussi l'alignement temporel des données à 1 ms près (formatage de données⁶).

3.4.1 Test d'acquisition statique

Ce test permet d'observer les biais présents dans les capteurs. Il faut donc examiner les sorties de ces capteurs quand l'entrée ne correspond à aucun mouvement i.e. translation et orientation fixes.

Protocole : le téléphone est posé sur une table et l'acquisition est lancée pendant 6 min (longue période).

Comportement attendu : les valeurs du gyroscope devraient être autour de 0 rad/s et garder la même variance tout au long du test, alors que l'accéléromètre présenterait 2 axes constants à 0 et un troisième qui serait constant mais à une valeur proche de $\pm 9.81 \text{ m/s}^2$ correspondant à la gravité.

6. <https://github.com/OSUPCVLab/mobile-ar-sensor-logger/wiki/Format-description#formats>

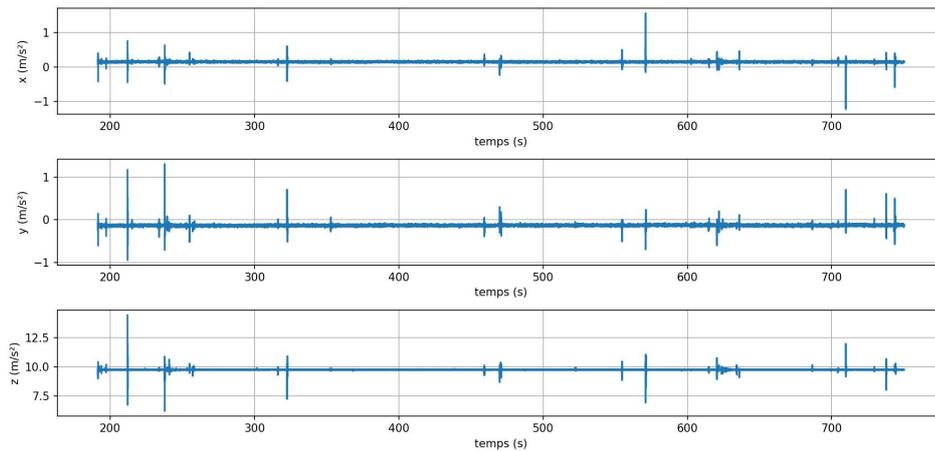


FIGURE 3.5 – Visualisation de la séquence statique de l'accélération.

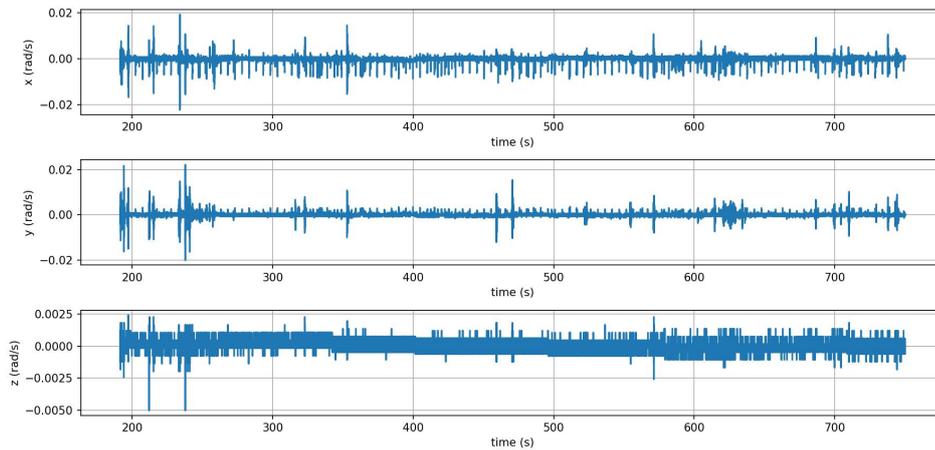


FIGURE 3.6 – Visualisation de la séquence statique de la vitesse angulaire.

Résultat : les valeurs obtenues en figures 3.5 et 3.6 sont cohérentes avec le comportement attendu.

Conclusion du test d'acquisition statique L'acquisition statique montre que les signaux sont bien stationnaires car ils ont des valeurs moyennes stables dans le temps et sont bruités additivement par un bruit blanc. . Il existe quand même une partie du vecteur gravité qui se retrouve sur les axes x et y de l'accéléromètre qui peut être liée à des effets de couplage d'axes ou positionnement du capteur. Certains pics sont présents dans les données. Ces pics sont des valeurs aberrantes et peuvent être liés des erreurs d'acquisition ou des erreurs d'écriture mémoires.

3.4.2 Calibration et non-calibration par Android

Dans la section 3.1.3.2, on a supposé qu'utiliser la calibration par l'algorithme "boite noire" de l'API d'Android était suffisant. Ce test compare l'acquisition statique via la même applica-

tion SensLogs (voir le tableau 3.1). Les synchronisations temporelle et spatiale de ces données sont garanties par l'application. On peut donc voir sur la figure 3.7 que les données calibrées semblent être centrées autour de 0 rad/s. Alors que les données non-calibrées présentent clairement des décalages aux origines, des biais.

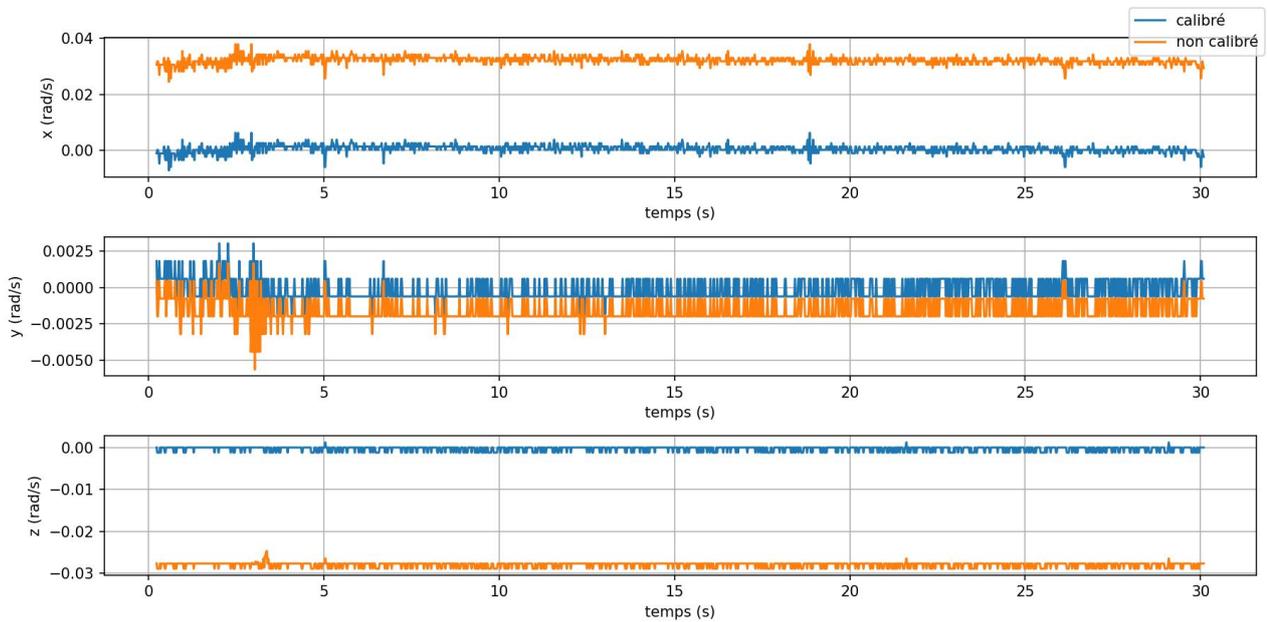


FIGURE 3.7 – Visualisation des mesures gyroscopiques calibrées par Android et non-calibrées.

Le tableau 3.3 montre les valeurs moyennes du gyroscope sur une fenêtre de 30 secondes pour chaque axe. Certains axes montrent que la calibration permet de réduire d'un facteur de 10, voir 50, le biais initial. Pour notre tâche, la calibration d'Android est jugée suffisante. Le test unitaire est validé.

acquisitions	moyenne axe x [rad/s]	moyenne axe y [rad/s]	moyenne axe z [rad/s]
calibrées	0.00069073	-0.0002091	-0.00096091
non calibrées	0.03230207	-0.0015835	-0.02860175

TABLE 3.3 – Biais des mesures gyroscopiques calibrées et non-calibrées.

Conclusion Les valeurs obtenues montrent que la calibration via l'algorithme Android est satisfaisante pour les données d'accéléromètre et du gyroscope.

3.4.3 Tests des translations unitaires

Dans chacun de ces tests unitaires, l'utilisateur met le téléphone sur une table. Il effectue un mouvement de translation demandé sur une distance connue. Le calcul de la double intégration de l'accéléromètre suivant chaque axe est affiché. Celui-ci doit correspondre à la valeur proche de celle que le protocole de test demande. Le test doit être répété pour vérifier s'il s'agit d'une erreur de protocole lié à un mauvais usage, ou d'une erreur systématique provenant des mesures du capteur de l'axe ou répartie sur plusieurs axes.

Dans la pratique, l'accélération est acquise par l'appel à une fonction de l'API d'Android avec une fréquence d'acquisition f_a . La vitesse et la position peuvent être obtenues par intégrale numérique, notée $cumsum()$, sur chaque axe via les équations (3.15) et (3.16).

$$\vec{a} = get_IMU(f_a) \tag{3.14}$$

$$\vec{v} = cumsum(\vec{a})/f_a \tag{3.15}$$

$$\vec{p} = cumsum(\vec{v})/f_a \tag{3.16}$$

Selon la documentation de l'API d'Android⁷, le vecteur de gravité sera présent dans les mesures. On devrait retrouver une valeur proche de $|\vec{g}| \approx \pm 9.81 \text{ m/s}^2$ sur un seul axe et/ou la norme d'accélération proche de cette valeur.

3.4.3.1 Translation sur l'axe x

Protocole : l'utilisateur pose le téléphone avec le port USB contre la table et le fait translater vers la droite (en direction du bouton verrouillé).

Comportement attendu : une distance de 60 cm est tracée sur la table. Le résultat de la double intégrale doit s'approcher de cette valeur sur l'axe x avec une erreur liée au cumul du bruit. L'axe y devrait être colinéaire au vecteur gravité.

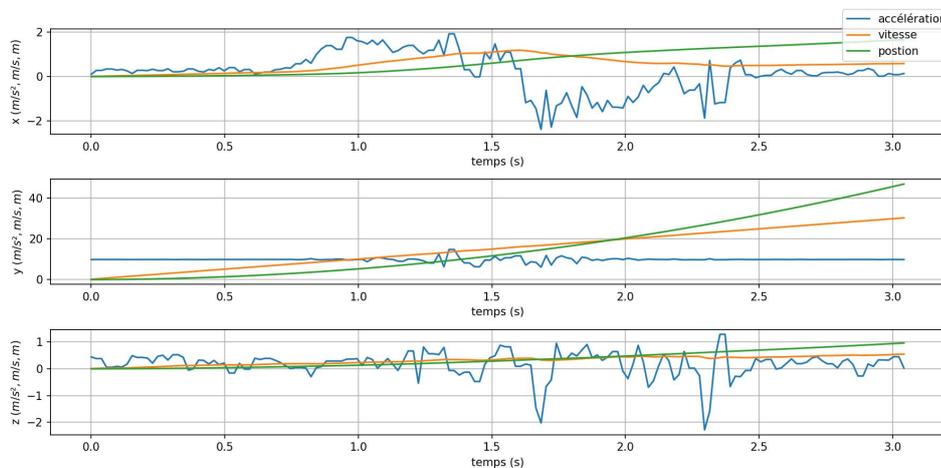


FIGURE 3.8 – Visualisation des données du test de translation sur l'axe x.

Résultat : la figure 3.8 montre que les allures de l'accélération, vitesse et position sont cohérentes sur l'axe x avec le mouvement demandé mais que les valeurs ne le sont pas.

L'axe y présente bien une accélération quasi-constante autour de 10 m/s^2 . Quand celle-ci n'est pas compensée, elle est accumulée par l'intégrale donnant un déplacement impossible de plus de 40 m.

L'axe z présente aussi un cumul sur la position même si le téléphone n'est pas sensé avoir translater sur cet axe. Ceci peut être lié à deux phénomènes : une composante de l'accélération

7. <https://developer.android.com/reference/android/hardware/SensorEvent#values>

est présente sur l'axe z et/ou l'utilisateur ne translate pas parfaitement le téléphone que sur l'axe x.

On observe alors sur la figure 3.9 l'intégrale de la norme de l'accélération compensée par la norme de gravité $|\vec{g}|$ comme dans l'équation (3.17). On a une translation de 0.43 m. Cela donne un écart de 0.17 m par rapport à la valeur ciblée.

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} - |\vec{g}| \quad (3.17)$$

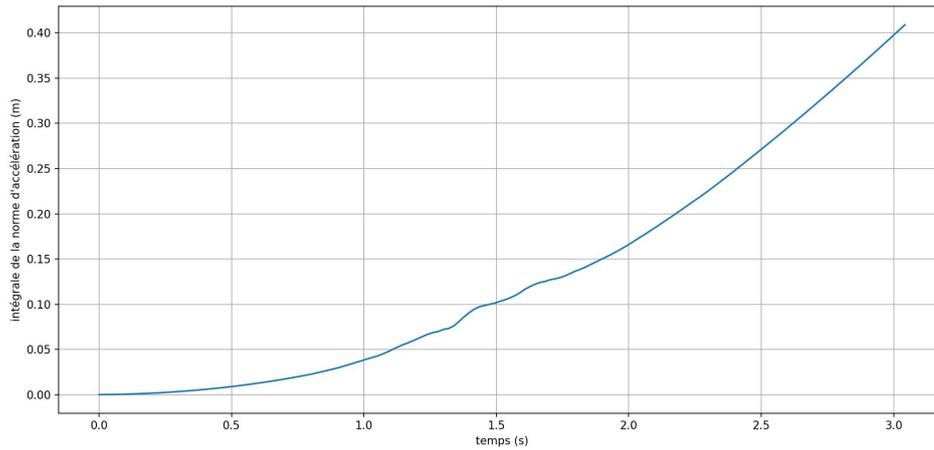


FIGURE 3.9 – Visualisation de la norme d'accélération du tests de translation sur l'axe x.

3.4.3.2 Translation sur l'axe y

Protocole : l'utilisateur pose le téléphone avec la caméra arrière sur la table. Il le translate vers le haut (au sens de la caméra avant).

Comportement attendu : une distance de 60 cm est tracée sur la table. Le résultat de la double intégrale doit s'approcher de cette valeur sur l'axe y avec une erreur liée au cumul du bruit. L'axe z devrait être colinéaire au vecteur gravité.

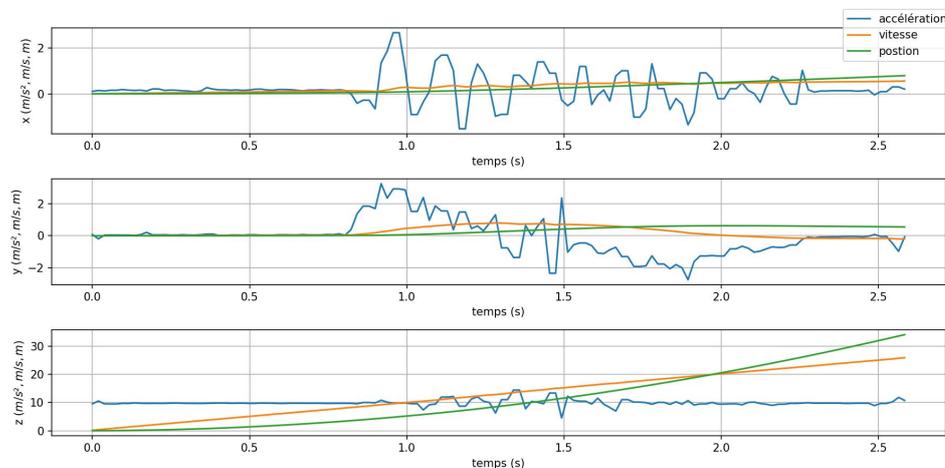


FIGURE 3.10 – Visualisation des données du test de translation sur l'axe y.

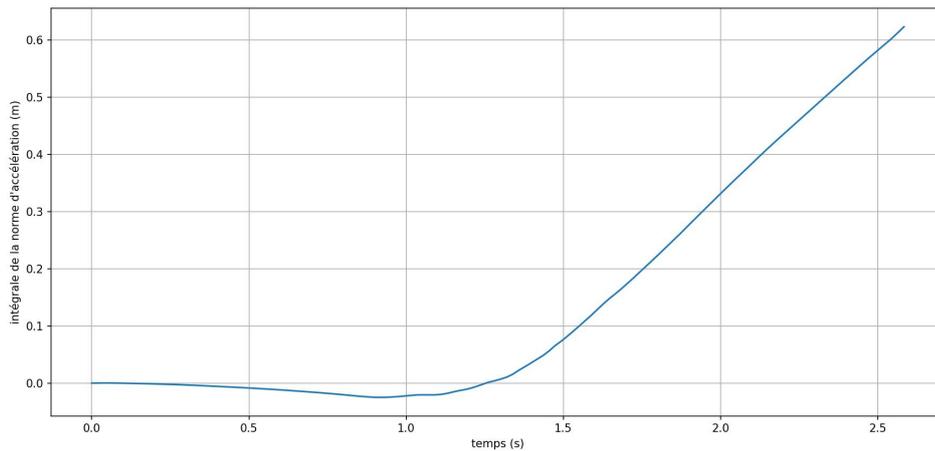


FIGURE 3.11 – Visualisation de la norme d'accélération du tests de translation sur l'axe y.

Résultat : la figure 3.10 montre que les allures de l'accélération, vitesse et position sont cohérentes sur l'axe y avec le mouvement demandé.

L'axe z présente bien une accélération quasi-constante autour de 10 m/s^2 . Quand celle-ci n'est pas compensée, elle est accumulée par l'intégrale donnant un déplacement impossible de plus de $\approx 32 \text{ m}$.

L'axe x présente aussi un cumul d'erreur sur la position même si le téléphone n'est pas sensé avoir translaté sur cet axe. Ceci peut être lié à deux phénomènes : une composante de l'accélération est présente sur l'axe x et/ou l'utilisateur ne translate pas parfaitement le téléphone que sur l'axe y.

On observe alors sur la figure 3.11 l'intégrale de la norme de l'accélération compensée par la norme de gravité $|\vec{g}|$ comme dans l'équation (3.17). La translation est de 0.62 m. Cela donne un écart de 0.02 m par rapport à la valeur ciblée.

3.4.3.3 Translation sur l'axe z

Protocole : l'utilisateur pose le téléphone en mode paysage sur la table et le fait translater droit devant (au sens de la caméra arrière).

Comportement attendu : une distance de 60 cm est tracée sur la table. Le résultat de la double intégrale doit s'approcher de cette valeur sur l'axe z avec une erreur liée au cumul du bruit. L'axe x devrait être colinéaire au vecteur gravité.

Résultat : la figure 3.12 montre que les allures de l'accélération, vitesse et position sont cohérentes sur l'axe z avec le mouvement demandé mais les valeurs sont relativement erronées (la vitesse finale est trop importante).

L'axe x présente bien une accélération quasi-constante autour de 10 m/s^2 . Quand celle si n'est pas compensée, elle est accumulée par l'intégrale donnant un déplacement impossible de plus de $\approx 33 \text{ m}$.

L'axe y présente aussi un léger cumul sur la position même si le téléphone n'est pas sensé avoir translaté sur cet axe.

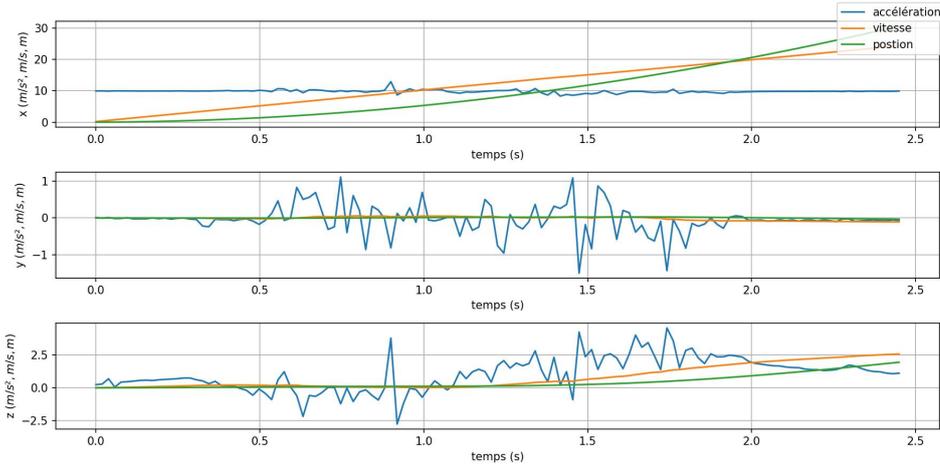


FIGURE 3.12 – Visualisation des données du test de translation sur l’axe z.

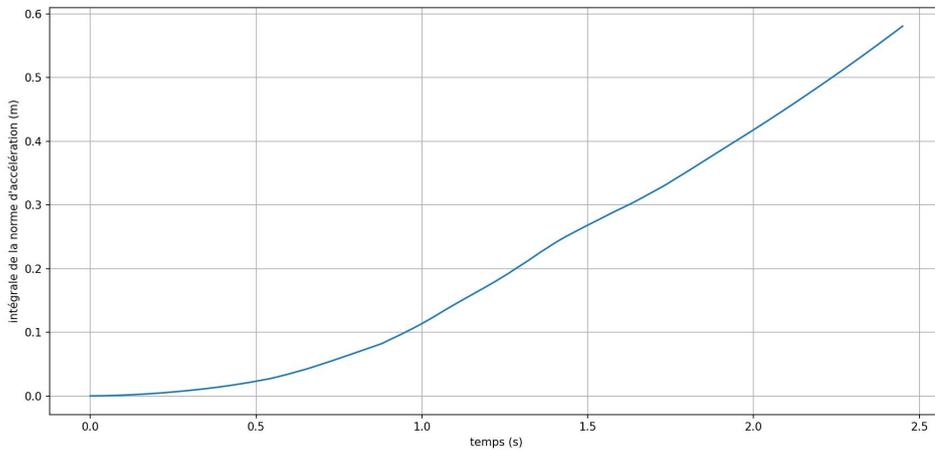


FIGURE 3.13 – Visualisation de la norme d’accélération du tests de translation sur l’axe z.

On observe alors sur la figure 3.13 l’intégrale de la norme de l’accélération compensée par la norme de gravité $|\vec{g}|$ comme dans l’équation (3.17). La translation finale est de 0.58 m. Cela donne un écart de 0.02 m par rapport à la valeur ciblée.

3.4.3.4 Conclusion des tests des translations unitaires

Nous notons que la double intégration des données brutes d’accélération est facilement erronée par la présence du bruit propre à l’accéléromètre même si l’allure des signaux reste correcte. Nous notons aussi que l’usage de la norme pour compenser la gravité permet d’avoir une meilleur estimation de la distance parcourue. Cette remarque fait écho à la différence d’usage entre la norme d’accélération et l’accélération verticale mentionnée en section 4.1.3.1.

3.4.4 Tests des rotations unitaires

De manière similaire aux tests unitaires d’accélération, dans chacun des tests unitaires, l’utilisateur pose le téléphone sur une table dans une orientation qui laisse l’axe à tester libre.

Il effectue ensuite un mouvement demandé. L'intégration du gyroscope suivant chaque axe est affichée. Celle-ci doit correspondre à la valeur que le protocole de test demande. Dans le cas où elle n'est pas conforme, le test doit être répété pour vérifier s'il s'agit d'une erreur de protocole liée à un mauvais usage, ou d'une erreur systématique provenant des mesures du capteur.

Naturellement, on demande un mouvement rotation de l'appareil qui est posé contre la table, il n'est pas possible d'avoir des mouvements strictement découplés entre les axes dû à l'imprécision de la manipulation humaine.

3.4.4.1 Rotation autour de l'axe x

Protocole : l'utilisateur fait tourner le haut de l'écran vers l'avant de 180° puis de -180° pour revenir à la même orientation.

Comportement attendu : l'intégration des données gyroscopiques devrait, sur l'axe x, commencer à 0° , atteindre $\pm 180^\circ$, puis revenir à 0° . Elle devrait être non nulle pour les axes y et z mais négligeable.

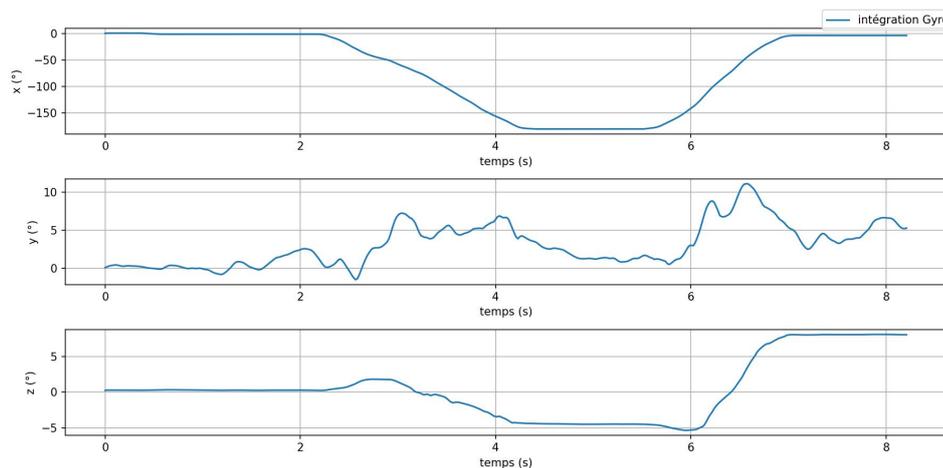


FIGURE 3.14 – Visualisation de la séquence de rotation autour de l'axe x du gyroscope.

Résultat : la figure 3.14 montre que l'allure des données gyroscopiques sur l'axe x est correcte et que la valeur du pic est à environ -170° . Les valeurs sur les axes y et z sont bien plus faibles mais non nulles.

3.4.4.2 Rotation autour de l'axe y

Protocole : l'utilisateur fait tourner le téléphone vers sa gauche de 180° puis de -180° pour revenir à la même orientation.

Comportement attendu : l'intégration des données gyroscopes devrait sur l'axe y commencer à 0° , atteindre $\pm 180^\circ$, puis revenir à 0° . Elle devrait être non nulle pour les axes x et z.

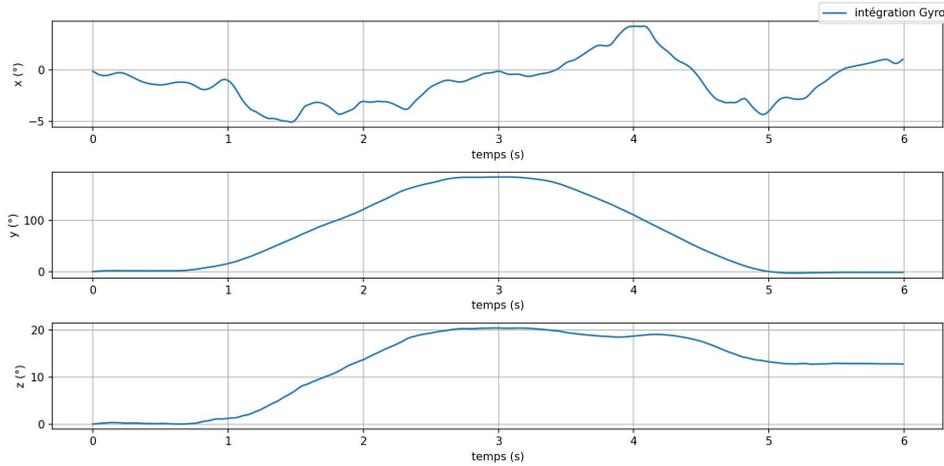


FIGURE 3.15 – Visualisation de la séquence de rotation autour de l’axe y du gyroscope.

Résultat : la figure 3.15 montre que l’allure des données gyroscopiques sur l’axe y est correcte. La valeur aux pics est autour de 180° . Ces valeurs sont proches de celles demandées par le protocole à la précision du gyroscope et du mouvement utilisateur. Le pseudo couplage entre les axes y et z est lié au fait que l’utilisateur n’a pas réussi à exactement produire le mouvement demandé avec le téléphone posé contre la table sur cet axe. Les valeurs sur les axes x et z sont correctes.

3.4.4.3 Rotation autour de l’axe z

Protocole : l’utilisateur fait tourner le téléphone de 180° autour de l’axe de la caméra arrière puis de -180° pour revenir à la même orientation.

Comportement attendu : l’intégration des données gyroscopes devrait sur l’axe z commencer à 0° , atteindre $\pm 180^\circ$, puis revenir à 0° . Elle devrait être non nulle pour les axes x et y.

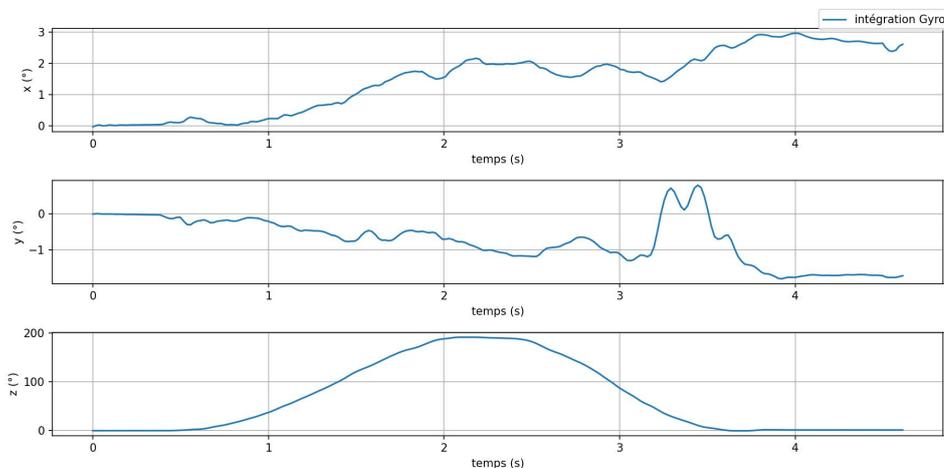


FIGURE 3.16 – Visualisation de la séquence de rotation autour de l’axe z du gyroscope.

Résultat : la figure 3.16 montre que l’allure des données gyroscopiques sur l’axe z est correcte. La valeur au pic est autour de 190° au lieu de 180° . Cette valeur est proche de celle demandée par le protocole à la précision du gyroscope et du mouvement utilisateur. Les valeurs sur les axes x et y sont correctes. Le test sur l’axe z possède les plus faibles variances sur les axes x et y car le téléphone est posé sur la table, écran vers le plafond. Puis, on effectue la rotation. L’axe z est quasiment tout le temps normal au plan de la table.

3.4.4.4 Conclusion des tests de rotations unitaires

De cette série de tests, nous notons que les axes du gyroscope sont suffisamment découplés entre eux. Les allures des valeurs des vitesses angulaires acquises sont correctes. Notons aussi une sensibilité plus importante à l’angle intégré qui devra être corrigé pour représenter correctement le cap du piéton.

3.5 Récapitulatif

L’acquisition de données inertielles sur téléphone est faite via une application qu’elle soit sous Android ou iOS. Les deux systèmes ont plusieurs couches d’abstraction qui éloignent l’utilisateur des capteurs embarqués. Il revient à l’utilisateur de choisir ou de programmer l’application d’acquisition selon les critères de son cahier des charges. En termes d’alignement spatial, les constructeurs garantissent un alignement satisfaisant et il est vérifié par nos tests. L’alignement temporel est garanti matériellement où chaque donnée de capteur est estampillée par la même horloge système.

La cadence de l’acquisition est unifiée sous celle du système du téléphone. Sachant qu’un humain se déplace à environ 2 Hz, [Michel et al., 2018] a estimé à 40 Hz la fréquence d’acquisition souhaitable. Notre téléphone *OnePlus 6T* offre 2, 10, 50 et 400 Hz. Nous utiliserons 50 Hz comme fréquence d’acquisition.

Les capteurs inertiels dans le téléphone sont embarqués sous la forme d’un MEMS aux dimensions inférieures à $100 \mu\text{m}$. La processus de fabrication de ces capteurs n’est pas parfait et plusieurs erreurs peuvent être retrouvées lors de l’acquisition.

Les erreurs liées au biais des capteurs sont corrigées avec la calibration. Nous avons examiné la différence entre les données issues de la calibration gyroscope et accéléromètre proposée par Android et les données non calibrées. La quasi-totalité des biais a été compensée par la calibration Android. Nous en avons conclu que la calibration Android était satisfaisante dans le cas où on n’utilise pas le magnétomètre, rejoignant la conclusion tirée par [Michel et al., 2018] sur le fait que le magnétomètre était recalibré à la volée sur Android et iOS. Donc, la calibration ne pose problème que pour le magnétomètre.

Les définitions des trois repères utilisés ont été présentées dans ce chapitre. Cela a servi à introduire les hypothèses sur le lien entre ces repères. Nous supposons que le repère piéton a la même verticalité que le repère bâtiment. Cette hypothèse semble cohérente car les piétons ne marchent pas en se penchant afin d’éviter le risque de perdre l’équilibre. Le téléphone quand à lui est tenu près du torse du piéton, ce qui verrouille ces angles d’orientation et les lie à ceux du piéton. Dans ce cas, seul le cap du téléphone, qui est le même pour le piéton, varie. Ces hypothèses sont valides dans le cas où le piéton possède cet équipement attaché au torse tel que les forces de l’ordre [Schneider, 2018] ou dans le cas où le cap peut être déduit par une estimation de pose effectuée en amont [Wang et al., 2021].

Les expériences menées ont pour but de valider le fonctionnement correct des capteurs embarqués dans le téléphone. Les protocoles établis et leurs résultats ont été présentés afin de valider un certain nombre de tests unitaires. Ces tests unitaires montrent l'impact de la calibration par Android pour une séquence d'acquisition statique. Les tests unitaires de translation montrent la présence du vecteur de gravité ainsi que les autres composantes de gravité dans chaque axe. L'erreur liée à l'intégration de ces accélérations étant importante, le besoin de déployer une technique pour estimer la longueur du pas est clair et est fait en section 4.1.

Les tests unitaires liés aux acquisitions de données angulaires montrent que les angles obtenus par intégration sont plus précis comparés aux résultats de l'intégration d'accélérations. Le problème restant consiste à déduire lequel des axes est l'angle de roulis, tangage ou cap. C'est pour cela que le filtre de *Madgwick* est déployé dans la section 4.1.4 en accord avec les hypothèses du positionnement du téléphone par rapport au corps de l'utilisateur.

Dans sa globalité ce chapitre permet de valider le processus d'acquisition comme une brique élémentaire fonctionnelle qui peut être intégrée au système que nous construisons.

3.6 Conclusion

Dans ce chapitre nous avons présenté la plateforme d'acquisition utilisée dans cette thèse. Les erreurs liées au processus d'acquisition ont été évaluées. Les définitions des différents repères nécessaires à notre étude ont été fournies. Ces repères permettent de poser les hypothèses considérées dans cette thèse. Enfin, une série de tests unitaires est présentée. Ces tests ont pour but de valider que les axes sont suffisamment découplés, mais la double intégration amène à des valeurs erronées sur la position. Sans cette validation, il serait impossible d'aborder le chapitre suivant qui concerne la prédiction du mouvement exploitant ces données d'accéléromètres et de gyroscopes.

Chapitre 4

Localisation par fusion de sources primaires

Sommaire

4.1	Navigation piétonne à l'estime (PDR)	64
4.1.1	Pré-traitement des données inertielles	64
4.1.2	Détection de pas	64
4.1.3	Estimation de la longueur de pas	66
4.1.4	Estimation du cap de pas	69
4.2	Métriques d'évaluation	70
4.2.1	<i>EvAAL framework</i>	71
4.2.2	Distances statistiques	72
4.2.3	Fonction de distribution cumulative (CDF)	73
4.2.4	Retour au départ	73
4.3	Vérité Terrain	73
4.4	Expérience : estimation d'une trajectoire	76
4.5	Récapitulatif	77
4.6	Conclusion	78

Dans ce chapitre, la navigation piétonne à l'estime (PDR) est présentée. Les processus de celles-ci sont : la détection de l'instant du pas, l'estimation de la longueur du pas et l'estimation du cap du pas. Les métriques d'évaluation pour les systèmes de localisation sont présentées. Une section présente les méthodes de mise en place de la vérité terrain. Une série de tests tirés d'expériences menées dans le but de valider le fonctionnement de cette navigation est présentée.

4.1 Navigation piétonne à l'estime (PDR)

L'approche nommée navigation piétonne à l'estime (Pedestrian Dead-Reckoning, PDR) est la plus utilisée quand le piéton est modélisé par un état discret. C'est une technique d'estimation du mouvement humain inspirée de la navigation à l'estime. Cette approche repose sur l'hypothèse que la position du piéton peut être calculée en mesurant son déplacement pas par pas et en accumulant la position précédente avec ce déplacement. Pour cela, elle passe par trois étapes : détection de pas ; calcul de la longueur et de la direction de chaque pas ; calcul de la nouvelle position.

La PDR requiert la présence d'une Unité de Mesures Inertielles (IMU) sur l'individu à localiser. Elle est soit portée à la main soit attachée à une autre partie du corps. Elle se sert de capteurs gyroscope et accéléromètre, qui fournissent les accélérations et les vitesses angulaires sur 3 axes pour prédire la distance parcourue et l'écart d'angle à chaque pas [Gu et al., 2019]. Pour un être humain, le déplacement correspond à une longueur de pas notée SL et à une variation d'angle de pas notée $d\theta$, calculées à des intervalles de temps discrétisés en fonction de la détection de l'occurrence d'un événement *pas*.

4.1.1 Pré-traitement des données inertielles

Les données inertielles sont acquises à une fréquence de 50 Hz via l'API du téléphone Android. Ces données sont bruitées par les différentes sources de bruits mentionnées dans le chapitre 3. Avant d'extraire l'information concernant le pas, il est nécessaire de les filtrer. Le filtrage de l'accélération est capital afin d'extraire les instants du pas. Les données du gyroscope sont filtrées par le filtre de *Madgwick*.

Les signaux d'accélération sont filtrés via un filtre passe bas avec une fréquence de coupure de 2 Hz. Car, cette fréquence correspond à un rythme de marche régulier d'un piéton. Une analyse proposée par [Susi et al., 2013] montre que pour les IMUs attachées au corps, 2 Hz est la fréquence qui englobe la majorité des marches. De plus, dans son analyse [Nicholas, 2022], *Nicholas* mentionne que l'athlète *Usain Bolt*, sur le segment de 20 m de l'exercice 60-80 m, a atteint la vitesse maximale de 12.42 m/s avec une fréquence de 4.4 pas par seconde. Dans la figure 4.1, une séquence d'accélération est filtrée sur chaque axe par un filtre de *Butterworth* pour cette fréquence de coupure.

4.1.2 Détection de pas

La détection de pas est une technique largement adoptée pour le modèle en représentation du piéton par état. Cette détection cadence la PDR du fait qu'elle crée des événements de pas. Quand ces événements occurrent, un traitement sur les données inertielles est effectué. Donc, au delà de l'instant du pas, cette détection définit aussi une fenêtre depuis laquelle les

données pour le traitement d'un pas sont considérées. Dans la figure 4.2, une visualisation de ce processus est montrée sur un pas.

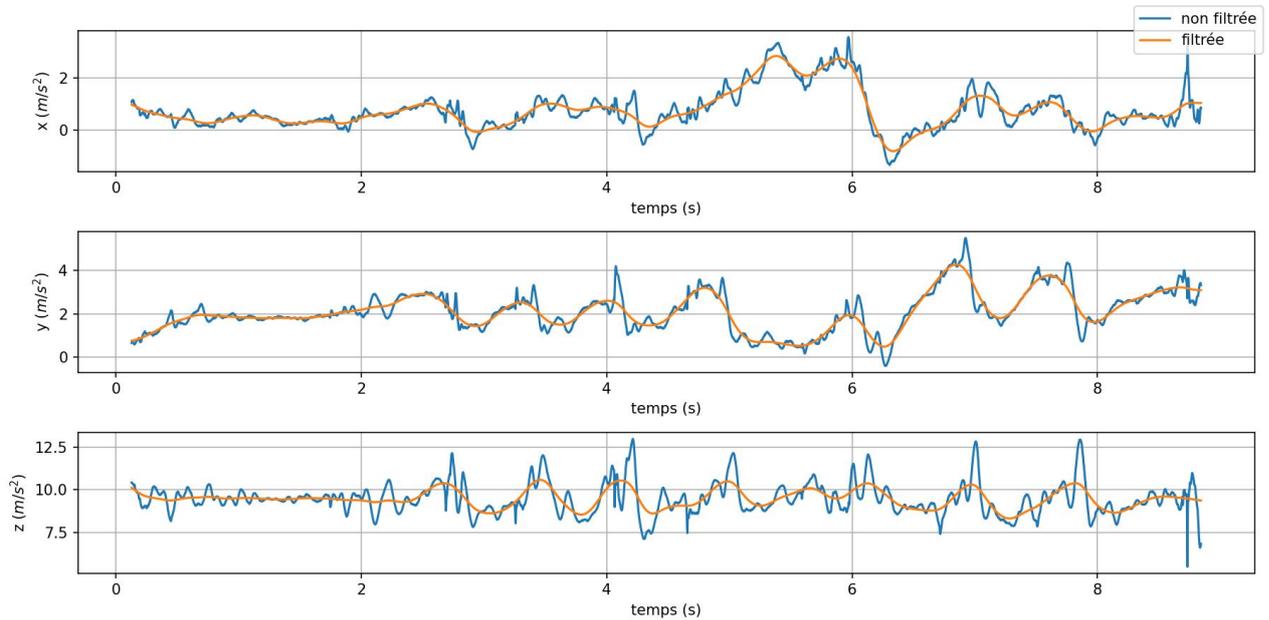


FIGURE 4.1 – Accélérations filtrées et non filtrées. Fréquence de coupure = 2 Hz.

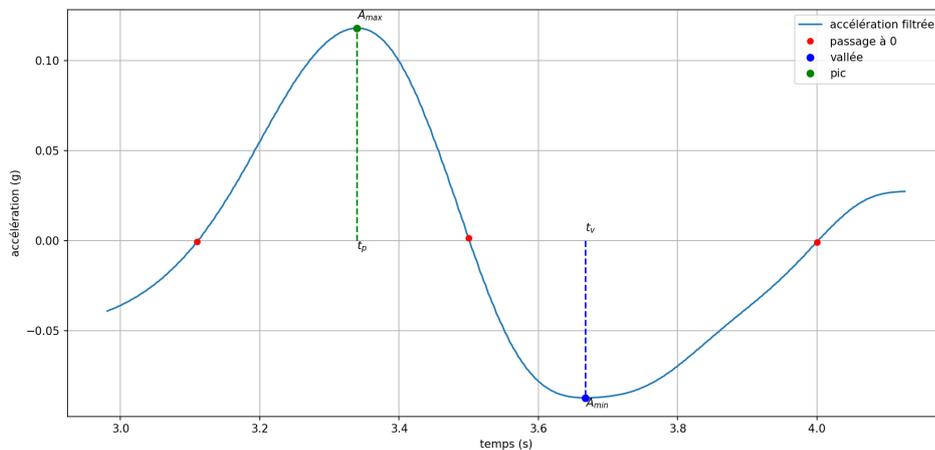


FIGURE 4.2 – Détection d'un pas visualisée.

Sur une fenêtre de temps, la norme de l'accélération $|\vec{A}|$ est centrée pour éliminer la composante de gravité. Ensuite, il faut chercher les instants du passage par 0 des valeurs de cette norme. Ces moments de passage du négatif/positif au positif/négatif définissent t^+/t^- . Le signal d'accélération d'un pas enchaîne obligatoirement la séquence : t^+, t^-, t^+ . Durant cet intervalle de temps, l'accélération atteint respectivement son maximum, minimum aux instants t_p, t_v suivant les équations (4.1) et (4.2).

$$t_p = \underset{t}{\operatorname{argmax}} |\vec{A}|(t) \quad | \quad t \in [t^+, t^-] \quad (4.1)$$

$$t_v = \underset{t}{\operatorname{argmin}} |\vec{A}|(t) \quad | \quad t \in [t^-, t^+] \quad (4.2)$$

Il est important de noter que le signal d'accélération peut effectuer plusieurs passages par 0 ce qui peut résulter en des faux positifs. Pour éliminer ces faux positifs, plusieurs conditions peuvent être stipulées, par exemple :

1. Placer un seuil, noté A_{th} , que le pic/vallée doit dépasser afin de le considérer comme candidat potentiel. [Khedr and El-Sheimy, 2017] ont proposé une méthode pour calculer ce seuil de manière adaptative.
2. S'assurer que la séquence : t^+, t^-, t^+ est cohérente dans le temps i.e. que t^+, t^- ne soit pas trop proches, ni trop éloignés. [Khedr and El-Sheimy, 2017] a implémenté cette condition pour s'assurer de la validé temporelle des pas détectés.

4.1.3 Estimation de la longueur de pas

Dans la section 2.2, il est mentionné que la PDR nécessite d'estimer la longueur de pas et qu'il existe plusieurs modèles pour effectuer cela.

4.1.3.1 Modèles paramétriques

Les modèles paramétriques sont basés sur des équations mathématiques qui supposent que la longueur du pas dépend du rythme de marche. Un premier modèle linéaire est proposé par [Alvarez et al., 2006] qui indique que la longueur de pas est définie par l'équation (4.3).

$$SL = K_1 f_{marche} + K_2 \quad (4.3)$$

Où f_{marche} est la fréquence de marche et K_1 et K_2 sont des paramètres liés à l'utilisateur déduits par une phase de pré-calibration. Une extension de ce modèle est proposée par [Shin et al., 2007] et récemment utilisée par [Ju et al., 2018] où il présume qu'il existe une corrélation linéaire entre la fréquence de marche et la variance des signaux de l'accéléromètre au cours d'un pas, et cela influence la longueur du pas. En conséquence, il détermine cette longueur en utilisant une combinaison linéaire de la fréquence de marche f_{marche} et de la variance de l'accéléromètre σ_a , comme indiqué dans équation (4.4)

$$SL = \alpha f_{marche} + \beta \sigma_a + \gamma \quad (4.4)$$

Où α, β et γ sont les paramètres liés à l'utilisateur déterminés eux aussi par une pré-calibration.

4.1.3.2 Modèle de pendule inverse

Ce modèle s'apparente aux modèles biomécaniques mais reste bien plus simple, [Cavagna et al., 1976] a suggéré que la marche consiste à déplacer le COM en pivotant autour du pied d'appui comme dans la figure 4.3. O_1 et O_2 sont le même point pivot à deux pas successifs. l est la hauteur du centre de masse (COM). γ_{min} et γ_{max} sont les angles d'écartement des jambes.

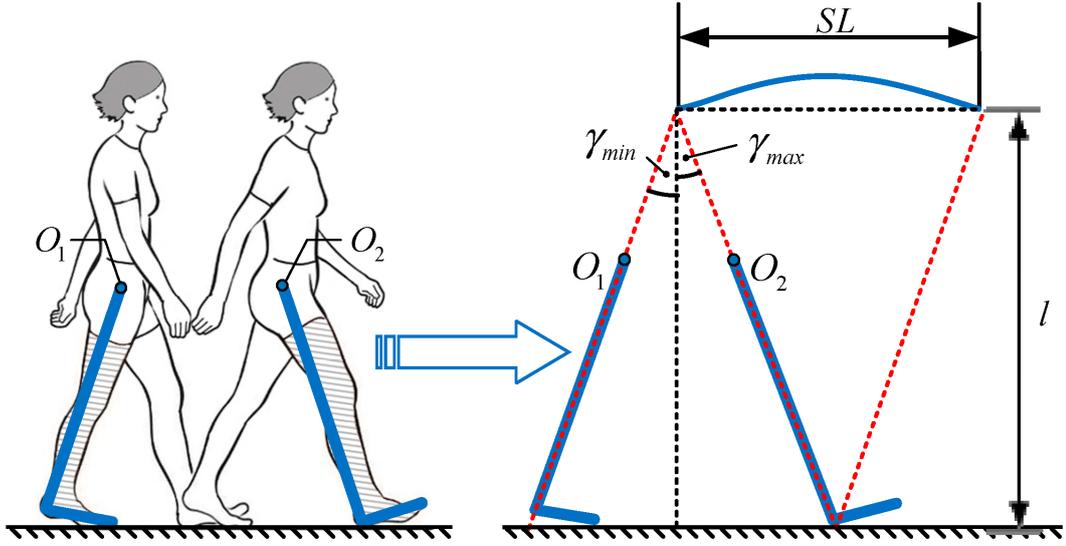


FIGURE 4.3 – Schéma représentant un pas adapté de [Zhao et al., 2019].

Plusieurs méthodes d'estimation de longueur de pas ont été proposées en se basant sur la relation entre le déplacement vertical du bassin et la longueur de pas [Díez et al., 2018]. Le calcul de cette longueur est défini par l'expression géométrique (4.5) proposée par [Zijlstra and Hof, 1997].

$$SL = K \times 2\sqrt{2LH - H^2} \quad (4.5)$$

Elle permet de relier la longueur du pas (SL), la longueur de la jambe (L), et le déplacement vertical du COM (H) pendant la phase simple support (voir figure 2.2), obtenu par double intégration de l'accélération verticale filtrée par un filtre passe-haut d'un capteur placé sur le bas du dos. Toutefois, ses auteurs ont constaté que ce modèle a tendance à sous-estimer la longueur du pas et qu'un paramètre d'étalonnage K est nécessaire pour ajuster les performances à chaque utilisateur.

4.1.3.3 Modèle empirique

Les modèles empiriques reposent sur le même principe de pendule inverse que les modèles paramétriques. Mais, au lieu de déduire la distance parcourue par un pas via l'équation (4.5), ces modèles proposent une approximation déduite d'expériences conduites sur des piétons. La méthode de *Harvey Weinberg* pour l'estimation de la longueur [Weinberg, 2002] est devenue une technique courante pour les applications de suivi de la condition physique et de la marche. Cette méthode est basée sur l'hypothèse selon laquelle la longueur de pas est linéairement corrélée avec la hauteur du COM pendant la marche. La hauteur du COM est mesurée à l'aide d'un accéléromètre attaché au niveau du bassin ou du torse, et la longueur de pas est calculée en multipliant cette hauteur par un coefficient de proportionnalité empirique suivant l'équation (4.6).

$$SL = K\sqrt[4]{a_{max} - a_{min}} \quad | \quad K \in \mathbb{R}_*^+ \quad (4.6)$$

où, a_{max} , a_{min} sont les pics/vallées durant un pas et K le gain empirique de *Weinberg* qui est propre à chaque piéton, proportionnel à la longueur de la jambe et est supposé constant. Il est important de noter que, dans son article d'origine [Weinberg, 2002], *Weinberg* indique que l'accélération a est l'accélération verticale relative au corps du piéton. Dans la figure 4.3, cette accélération est orientée dans le même sens que l (verticale). Pourtant, la méthode de *Weinberg* a aussi été testée [Jimenez et al., 2009] en utilisant la norme de l'accélération calculée via l'équation (3.17) avec le vecteur de gravité compensé. Il s'avère que la différence entre utiliser a et a_z est négligeable.

Une autre approche a été proposée par [Kim et al., 2004] qui est décrite par l'équation (4.7).

$$SL = K \times \sqrt[3]{\frac{\sum_{i=1}^W |a_i|}{W}} \quad (4.7)$$

Où W est la fenêtre temporelle des échantillons considérés et a_i l'accélération aux instants i dans cette fenêtre. K est le gain propre à l'utilisateur qui doit être pré-calibré.

Les auteurs [Scarlett, 2007] ont proposé une méthode qui lie empiriquement l'intensité moyenne des pics d'accélération normalisés par la différence entre l'accélération maximale et minimale dans la fenêtre temporelle W dans l'équation (4.8) où K est le gain qui doit être pré-calibré. Cette expression a pour but d'éliminer l'effet ressort présent dans la marche [Hou and Bergmann, 2020b].

$$SL = K \times \frac{\sum_{i=1}^W |a_i|/W - a_{min}}{a_{max} - a_{min}} \quad (4.8)$$

Dans l'article [Hou and Bergmann, 2020a], les auteurs ont effectué un comparatif entre les modèles de *Weinberg*, *Kim*, *Scarlett*, *Shin* et paramétrique linéaire en mesurant une distance à parcourir de 11.28 m. Une seule acquisition est faite pour cette distance dont les données inertielles sont utilisées par ces différentes méthodes. Le tableau 4.1 qui montre que les méthodes de *Kim*, *Weinberg* et *Shin* ont les plus petites erreurs absolues et relatives.

Méthode	linaire	Weinberg	Kim	Scarlett	Shin
Distance estimée [m]	8.86	11.11	11.23	10.08	11.12
Erreur [m]	2.41	0.16	0.05	1.20	0.15
Taux d'erreur [%]	21.42	1.48	0.45	10.67	1.41

TABLE 4.1 – Résultats de la comparaison faite par [Hou and Bergmann, 2020a].

De cette étude, on peut conclure que les méthodes de *Weinberg*, *Kim* et *Shin* sont préférables aux 2 autres. La méthode de *Shin* demande de déterminer 3 paramètres. La méthode de *Kim* est plus lente en temps de calcul selon [Hou and Bergmann, 2020a]. L'avantage de la méthode de *Weinberg* est qu'elle ne demande qu'un seul paramètre utilisateur. Ce qui fait qu'une plus grande flexibilité sur le contrôle de ce paramètre est possible pour corriger l'estimation de la longueur du pas. Le choix de la méthode de *Weinberg* nous permet la correction de cette estimation dans le chapitre 5. Notre étude de l'impact du gain de *Weinberg* sur les trajectoires est présentée dans la section 5.5.2 et a été publiée dans [Ghaoui et al., 2022]. Les recherches plus récentes ont utilisé la méthode *Weinberg* [Deng et al., 2015, Yan et al., 2018, Luo et al., 2020a, Ghaoui et al., 2022]. Des méthodes pour estimer plus finement ce paramètre ont été proposées.

Par exemple, [Strozzi et al., 2018] donne l'équation (4.9) pour estimer ce gain. Leur approche adopte un gain constant par utilisateur pour l'estimation de la longueur du pas qui repose sur la norme d'accélération comme "indicateur de quantité de mouvement". Cette formule leur permet de calibrer le gain du déplacement où β est une constante estimée par utilisateur. Cette méthode introduit quand même un paramètre par utilisateur supposé constant.

$$K = \beta \times \frac{1}{\sqrt[3]{a_{max}(i)}} \quad (4.9)$$

4.1.4 Estimation du cap de pas

Dans le chapitre 2, section 2.2.3, plusieurs techniques d'estimation du cap du piéton sont présentées. Le filtre de [Madgwick et al., 2011] utilise une représentation de quaternion, permettant aux données d'accéléromètre et de magnétomètre d'être utilisées dans un algorithme de descente de gradient analytiquement dérivé et optimisé pour calculer la direction de l'erreur de mesure du gyroscope en tant que dérivée de quaternion. Il existe aussi dans une version sans données magnétométriques. Celle-ci est retenue pour notre étude car elle permet de suivre l'évolution des angles d'orientation du téléphone avec uniquement les données accélérométriques et gyroscopiques.

Dans la figure 4.4, les angles d'orientation du téléphone : roulis ϕ , tangage ψ et cap α sont représentés. Le filtre de *Madgwick* permet d'estimer les angles ϕ et ψ dans le temps grâce au vecteur de gravité présent dans les données d'accélération.

Donc, seul l'angle α reste relatif car sans magnétomètre le filtre de *Madgwick* procède à une simple intégrale numérique du signal de vitesse angulaire de cet angle. Même, si le calcul des angles d'orientation est fait via des quaternions par le filtre, le résultat est converti en angles d'Euler pour exploiter l'information du cap.

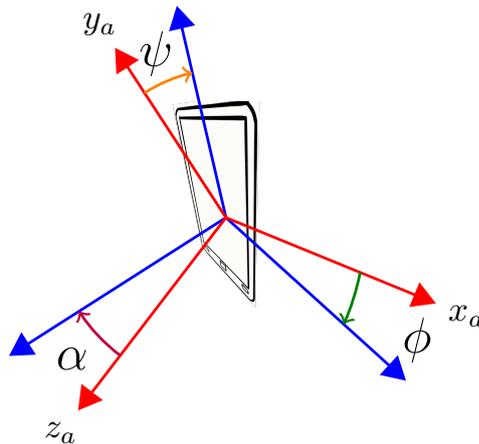


FIGURE 4.4 – Visualisation des 3 axes du gyroscope d'un téléphone.

L'équation (4.10) montre l'intégrale numérique des vitesses angulaires ω sur chaque axe et la fréquence d'acquisition f_a . Dans la figure 4.5, une comparaison entre l'intégration numérique des données est comparée à la sortie du filtre *Madgwick* convertie aux angles d'Euler.

$$\begin{bmatrix} \psi \\ \phi \\ \alpha \end{bmatrix} = \text{cumsum} \left(\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} / f_a \right) \quad (4.10)$$

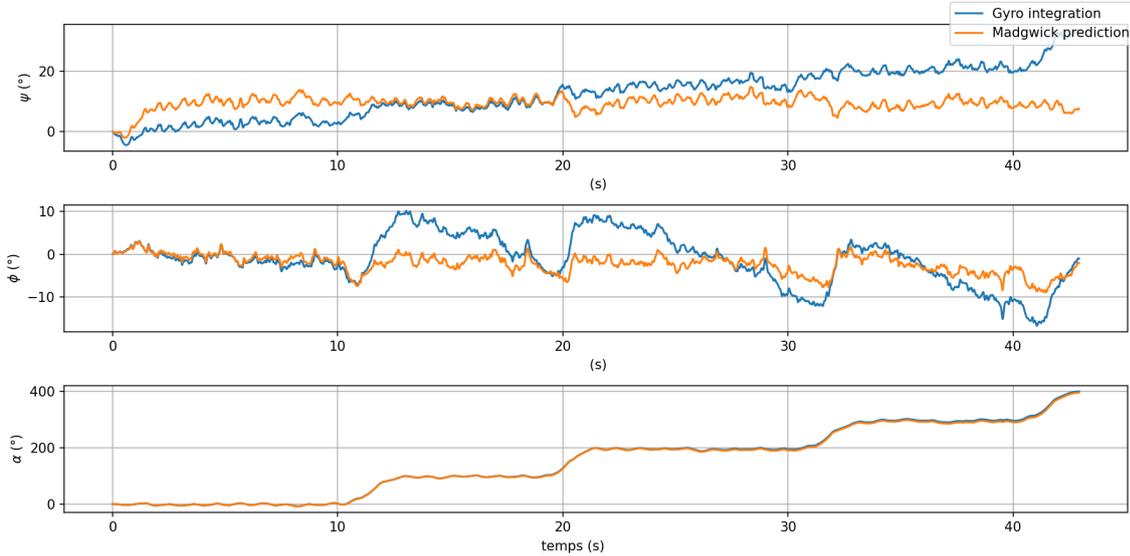


FIGURE 4.5 – Comparaison entre l'évolution des angles ϕ , ψ et α .

Le téléphone est maintenu dans la même orientation pendant que l'utilisateur se déplace. Ce n'est pas la trajectoire qui est examinée mais l'évolution des angles. Les angles ψ et ϕ par intégration montrent la présence d'une dérive sur ces deux angles. Cette dérive est liée à l'erreur cumulée vue en section 3.1. Il existe aussi un couplage entre les angles ψ et ϕ .

Le filtre de *Madgwick*, arrive à réguler cette erreur en exploitant la présence du vecteur de gravité dans les données d'accélération. D'où le fait que les valeurs moyennes de ces deux angles soient relativement stables dans le temps.

Pour l'angle α , le filtre de *Madgwick* et l'intégration du gyroscope ont la même allure mais une différence de valeur. Cette différence est négligeable comparée à l'ordre de grandeur des virages pris durant le déplacement. Mais, plus la trajectoire est longue, plus ces erreurs causent une dérive importante. Nous devons proposer une méthode de correction du cap pour réguler l'accumulation de l'erreur gyroscopique. Cette méthode exploite la carte du bâtiment et sera présentée dans la section section 5.3.

4.2 Métriques d'évaluation

Les critères pour juger des performances de ces systèmes de localisation sont divers et peuvent varier selon l'application visée. L'aspect statistique définit la précision d'un système comme l'erreur entre la position estimée et la position réelle ainsi que l'incertitude autour de cette position estimée. On peut aussi considérer le maximum que cette erreur peut atteindre comme étant un critère d'acceptabilité. À l'acceptabilité, on associe souvent la robustesse car tous les signaux et informations utilisés par ces systèmes sont bruités et peuvent donc être erronés et donc moins un système est sensible à ces perturbations, plus il est robuste. La

finesse de localisation d'un système se décline en différents niveaux : à l'étage, à la salle, à la chaise près, à la position (x, y) .

4.2.1 *EvAAL framework*

Le *framework EvAAL* [Potortì et al., 2017] a été conçu dans le but d'évaluer les systèmes *Ambient assisted living* (AAL) [Dobre et al., 2017] grâce à une analyse comparative concurrentielle [Potortì et al., 2015]. L'idée est de rassembler des systèmes fonctionnels, à la fois prototypes et matures, et de comparer indépendamment leurs performances dans un ou plusieurs domaines spécifiques, avec l'objectif à long terme de créer un ensemble de référentiels d'évaluation pour les systèmes en intérieur [Barsocchi et al., 2012, Potortì et al., 2015, Barsocchi et al., 2013]. *EvAAL* est utilisé dans les conférences IPIN depuis 2014.

Le *framework* considère les points suivants comme clefs dans la comparaison des systèmes de localisation :

1. Mouvement naturel d'un acteur : l'agent testant un système se déplace spontanément en suivant un scénario prédéfini, incluant la marche, le repos ou d'autres actions, pour simuler un mouvement naturel dans un environnement réaliste.
2. Environnement réaliste : la scène dans laquelle les tests sont effectués doit être un décor réaliste, pour garantir une évaluation fidèle des performances du système de localisation.
3. Résolution de mesure réaliste : les erreurs minimales de temps et d'espace sont déterminées en fonction du mouvement de l'acteur. La résolution spatiale est basée sur le diamètre de la projection du corps de l'acteur au sol, fixé à 50 cm. La résolution temporelle est définie par le temps qu'une personne met pour parcourir une distance égale à la résolution spatiale. Dans un environnement intérieur, en considérant une vitesse maximale de 1 m/s, la résolution temporelle est de 0,5 s.
4. Troisième quartile de l'erreur euclidienne ponctuelle : le score de précision est basé sur le troisième quartile de l'erreur, qui est calculé comme la distance euclidienne 2D entre les points de mesure et les points estimés.

Au-delà de ceux-ci, les critères étendus du cadre *EvAAL* incluent les éléments suivants :

1. Chemin secret : le chemin final à parcourir est divulgué uniquement au concurrent dont le système est en cours de test, juste avant le début du test, pour éviter que les concurrents conçoivent des systèmes exploitant des caractéristiques spécifiques du chemin.
2. Acteur indépendant : l'acteur qui se déplace dans l'environnement de test est un agent non formé à l'utilisation du système de localisation, pour garantir une évaluation réaliste et impartiale du système.
3. Système d'enregistrement indépendant : le système du concurrent estime la position deux fois par seconde et envoie les estimations à l'application fournie par le comité de test. Cela empêche toute action malveillante de la part des concurrents. Le système d'enregistrement est un logiciel libre.
4. Parcours et timing identiques : l'acteur parcourt le même parcours identique avec le même timing identique pour tous les concurrents, dans les limites d'erreurs de temps et d'espace définies selon les résolutions spécifiées ci-dessus, pour garantir des conditions de test équitables entre les concurrents.

4.2.2 Distances statistiques

Il est possible de mesurer la précision d'un système de localisation produisant une position $P = (x, y)$ en calculant la distance euclidienne E en mètres entre cette position et la position d'un point de repère notée $P_{ref} = (x_{ref}, y_{ref})$ comme dans l'équation (4.11).

$$E = \sqrt{(x - x_{ref})^2 + (y - y_{ref})^2} \quad (4.11)$$

Si la sortie du système n'est pas ponctuelle mais consiste plutôt en une distribution de points \mathcal{P} étant vue comme des positions possibles, l'erreur euclidienne est mesurée dans ce cas entre la position moyenne $\bar{P} = (\bar{x}, \bar{y})$ des sorties possibles et la position du point repère.

$$\bar{P} = \left[\sum_{i=1}^N x_i / N \quad , \quad \sum_{i=1}^N y_i / N \right] \quad (4.12)$$

La distribution des positions P permet de calculer une autre grandeur statistique : la distance de Mahalanobis. Cette distance est une mesure de la distance entre un point repère et une distribution statistique dans un espace multidimensionnel. Elle tient compte de la covariance entre les différentes dimensions de l'espace, ce qui permet de prendre en compte les corrélations entre les variables. Elle est souvent utilisée pour comparer un point à une distribution multivariée, en prenant en compte la structure de covariance de cette distribution. Pour une distribution \mathcal{P} de points P_i de moyenne \bar{P} , la distance de Mahalanobis est donnée par équation (4.13).

$$M = (P_{ref} - \bar{P}) \times \mathbf{S}^{-1} \times (P_{ref} - \bar{P})^T \quad (4.13)$$

Avec $\mathbf{S} \in \mathbb{R}^{2 \times 2}$ étant la matrice de covariance de \mathcal{P} définie par l'équation (4.14) :

$$\mathbf{S} = \mathbb{E}_i \left[(P_i - \bar{P}) \times (P_i - \bar{P})^T \right] \quad (4.14)$$

Dans la section 5.2.3, nous présentons cette distribution \mathcal{P} mais dans le cas où ses membres ont des importances différentes. i.e. un poids w_i est assigné à chaque position. Ces poids respectent la contrainte de l'équation (4.15).

$$\sum_{i=1}^N w_i = 1 \quad (4.15)$$

Dans ce cas la position moyenne est calculée via la moyenne pondérée :

$$\bar{P} = [\bar{x}, \quad \bar{y}] = \left[\sum_{i=1}^N w_i x_i \quad , \quad \sum_{i=1}^N w_i y_i \right] \quad (4.16)$$

Et l'équation (4.14) devient l'équation (4.17).

$$\mathbf{S} = \mathbb{E}_i \left[w_i (P_i - \bar{P}) \times (P_i - \bar{P})^T \right] \times v \quad (4.17)$$

Où v est le terme de normalisation défini par :

$$v = \left(\left(\sum_{i=1}^N w_i \right)^2 - \sum_{i=1}^N w_i^2 \right)^{-1} \quad (4.18)$$

Enfin, dans un scénario réaliste, la vérité terrain n'est disponible que sur un ensemble de points repères qui sont identifiés par un indice t . Il suffit donc de calculer les erreurs euclidiennes et Mahalanobis moyennes sur les K points repères et utiliser l'équation (4.19) pour quantifier les erreurs sur la trajectoire complète.

$$E = \sum_{t=1}^K E_t / K \quad , \quad M = \sum_{t=1}^K M_t / K \quad (4.19)$$

4.2.3 Fonction de distribution cumulative (CDF)

La fonction de distribution cumulative (CDF) est une fonction statistique qui permet de décrire la probabilité qu'une variable aléatoire prenne une valeur inférieure ou égale à une certaine valeur donnée. Elle est souvent utilisée pour représenter la distribution des valeurs d'une variable aléatoire et pour calculer des probabilités associées à cette distribution.

Mathématiquement, la fonction de distribution cumulative d'une variable aléatoire X est définie par :

$$F(x) = P(X \leq x) \quad (4.20)$$

Où x est une valeur réelle et $P(X \leq x)$ est la probabilité que la variable aléatoire X prenne une valeur inférieure ou égale à x . La CDF peut être utilisée pour calculer plusieurs statistiques importantes, telles que la médiane ($F(x) = 0,5$), les quartiles ($F(x) = 0,25$ et $F(x) = 0,75$), et d'autres mesures de tendance centrale et de dispersion.

Pour la conférence IPIN [Potortì, 2020], elle représente le moyen de classer les différents systèmes entre eux via le *framework EvAAL*. Ceci est fait en calculant la CDF à 75% des erreurs. Dans la pratique, les erreurs euclidiennes à chaque point repère sont calculées. Elles sont ensuite triées dans l'ordre croissant. Puis, l'histogramme cumulé normalisé de ces erreurs est calculé. Enfin, la valeur de l'erreur incluant 75% de l'histogramme est retenu comme CDF-75.

4.2.4 Retour au départ

L'erreur de retour au départ est une mesure ponctuelle permettant d'indiquer si un utilisateur a accompli une trajectoire qui se termine à la même position dans laquelle elle a commencé. Cette mesure permet de rapidement identifier si la localisation a échoué ou que le piéton est perdu. [Zhao et al., 2019] utilise ce critère pour montrer l'amélioration apportée par leur système de localisation. Dans [Ghaoui et al., 2022], nous avons considéré 2.5 m comme étant l'erreur maximale de retour au départ ce qui correspond à cinq fois la résolution spatiale définie par l'*EvAAL*.

4.3 Vérité Terrain

Afin d'effectuer les mesures nécessaires sur les trajectoires et déplacements effectués par le piéton, il est nécessaire de connaître sa position et son orientation à l'instant de mesure. Établir la vérité terrain dans l'environnement intérieur n'est pas toujours possible à cause de la précision demandée. Des solutions qui s'en approchent existent. Ces solutions varient en

résolution spatiales et temporelles, en précision de mesure, en erreur maximale tolérée et en coût de déploiement. Le choix de la solution est un compromis qui est fait en accord avec la précision exigée par l'application visée. Dans notre étude, l'application requiert une précision de l'ordre de 10 *cm*.

Tracé manuel du chemin. Ce tracé est destiné à guider les piétons sur une trajectoire spécifique. Il peut également servir de vérité en étant divisé uniformément en plusieurs points discrets. Ces points peuvent être utilisés pour comparer la trajectoire estimée à ce tracé. Bien que cette méthode soit peu coûteuse, elle n'est pas très précise ni fidèle à la réalité, car la trajectoire réelle du piéton ne correspond pas forcément à la vérité terrain.

Empreintes de peinture, de craie ou de poussière. Les pas étant discrets, il est possible de voir l'empreinte laissée par un pied au sol. Ce qui fait que en termes de fidélité de mesures, cette empreinte est exacte. En utilisant de la peinture sur la semelle du piéton par exemple, la position et l'orientation de chaque pas est enregistré sur directement sur le sol. Il suffit alors de collecter ces empreintes. Cette vérité terrain fait face à plusieurs inconvénients : elle requiert un temps de préparation et déploiement conséquent et elle n'est pas toujours autorisée à être effectuée car elle peut dégrader l'espace où l'expérience est menée. Un moyen supplémentaire de collection de position/orientation spatiale doit être déployé pour chaque pas. L'erreur de mesure de ce moyen doit être bien plus faible que le système de localisation testé. Enfin, Cette méthode ne donne que la vérité terrain spatiale et l'ordre temporel dans lequel les pas s'enchaînent.

GNSS est une technologie mature qui a atteint des niveaux de précision dépendants du nombre de satellites vus par le récepteur. Les systèmes Real Time Kinetic (RTK) peuvent être plus précis en positionnant un récepteur statique (*Base*) et un récepteur mobile (*Rover*). Le problème avec les signaux GNSS est qu'ils ne sont pas exploitables à l'intérieur d'un bâtiment à cause des bâtiments bloquant le signal émis par le satellite.

Radio-localisation. Dans le chapitre 2 section 2.2.3, plusieurs techniques utilisant les signaux radiofréquences sont présentées. L'utilisation de ces systèmes est possible si l'erreur maximale tolérée de la vérité terrain est bien plus faible que l'erreur du système de localisation. Cette vérité terrain fait face à plusieurs difficultés :

- L'usage d'une technique exploitant les technologies de communication réseau nécessite que ce réseau soit mis en place s'il n'existe pas ou de s'assurer qu'il couvre les besoins en résolutions spatiale et temporelle s'il existe déjà. Dans les deux cas, il existe un coût non négligeable pour rendre la vérité terrain opérationnel.
- Les systèmes de localisation et réseau ont des sources temporelles différentes. Il est donc nécessaire de produire une synchronisation temporelle en ligne ou hors ligne.

Localisation et cartographie simultanées (SLAM) est une technique qui enchaîne itérativement deux étapes : la localisation et la cartographie. Un système SLAM se localise via une estimation de position. Ensuite, la cartographie de l'environnement est construite à partir de caractéristiques présentes dans les signaux collectés. Le but est qu'en enchaînant la localisation et la cartographie, le système puisse affiner la position et la carte. Il existe un

mécanisme de fermeture de boucles permettant de détecter les zones de la carte construite qui sont revisitées et de recalculer la trajectoire sur ces passages.

Le SLAM demande un effort d'implantation important. Malgré les mécanismes existants pour renforcer ses deux processus, il reste sensible aux fausses détections de caractéristiques causées par des environnements dynamiques. Cela peut compromettre le système et le rendre incohérent. Il n'est donc acceptable que pour établir une pseudo-vérité terrain.

Semelle connectée permet de détecter le contact du pied avec le sol comme le modèle de [Binelli et al., 2023]. Une IMU attachée au pied peut exploiter l'approche *Strap Down* et détecter les moments où la vitesse est nulle pour estimer la distance parcourue [Renaudin, 2018]. Cette approche pour établir la vérité terrain nécessite le port des capteurs sur le pied et que la synchronisation avec le système de localisation soit effectuée. La mesure la plus précise que peut donner cette semelle connectée est l'instant de détection des pas.

Système de capture du mouvement est déployé par le cinéma pour animer un avatar virtuel par des mouvements d'acteur humain. Cette tâche exige une précision très élevée que ce système assure : 1 cm d'erreur en position et 0.5° d'erreur en orientation. Cette précision a un coût très important car ces systèmes déploient plusieurs caméras (> 20). Ils sont donc limités à une grande salle et ne peuvent pas couvrir un bâtiment en entier.

Réalité Augmentée (AR) et Réalité Virtuelle (VR). Les systèmes VR utilisent une solution clef en main où l'utilisateur porte un casque de réalité virtuelle qui est localisé par des balises statiques préposées. Cela implique donc que la zone où le système peut fournir la vérité terrain est bornée par ces balises statiques. Même si le coût de déploiement de ces balises est moyen, l'impossibilité de mise à l'échelle au niveau d'un bâtiment rend cette option insatisfaisante. Pour les systèmes AR, qu'ils nécessitent ou non la pose d'amers artificiels ou l'identification d'amers remarquables dans le bâtiment, ils fournissent une pseudo-vérité terrain dont l'erreur est du même ordre de grandeur que celle du système de localisation testé.

Tapis connectés sont déployables sur la surface où l'expérience est conduite. Chaque tapis forme une matrice de capteurs de pression qui au contact du pied enregistre l'information de temps, position et/ou orientation. Chaque tapis peut donc fournir la vérité terrain des pas effectués par le piéton sur celui-ci. En termes de coût, la résolution spatiale dépend de la densité en capteurs de la matrice de pression et de la surface à couvrir. Deux opérations d'alignement sont nécessaires : temporel si les mesures de chaque tapis est estampillée par sa propre horloge ; spatial car les tapis rectangulaires doivent être disposés afin de couvrir une surface de géométrie arbitraire.

Points repères sont des points dans le bâtiment dont la position est calibrée avec un télémètre laser. Ils sont discrets et distingués par un symbole posé au sol : une croix dans notre cas. Durant une expérience, il est demandé au piéton de suivre le chemin des croix et de poser le pied sur celles-ci. Il est possible de signaler le passage sur une croix via un bouton d'estampillage dans le système d'acquisition. Dans le téléphone, l'application d'acquisition permet cette action. Cette estampille est donc alignée temporellement avec les données acquises durant l'expérience. Cette vérité terrain est simple à établir, possède un coût de déploiement négligeable, préserve l'environnement d'expérience, présente une erreur de l'ordre du pas.

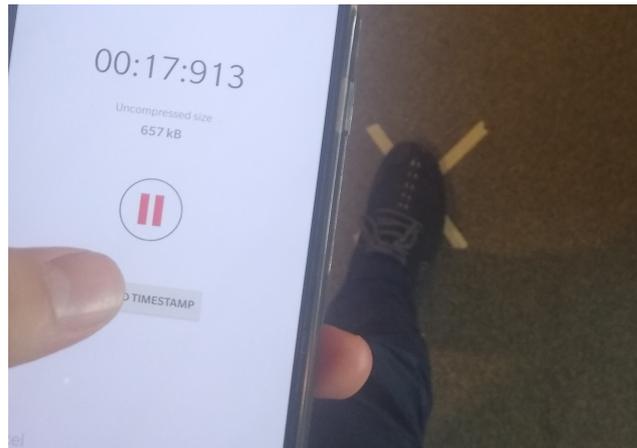


FIGURE 4.6 – Utilisateur déclarant son passage sur un point repère.

Cette erreur est liée au fait que l'utilisateur appuie sur le bouton d'estampillage un pas en avance ou en retard. Dans nos tests, ce phénomène ne s'est pas présenté mais il reste une possibilité. De plus, il est possible de vérifier si l'appui correspond au moment où le piéton marche sur la croix via la vidéo acquise. C'est pour ces raisons que nous avons donc adopté cette méthode pour réaliser nos expériences de manière similaire aux conférences IPIN [Potorti, 2020]. Dans la figure 4.6, une visualisation du moment de passage du piéton sur la croix est montrée.

4.4 Expérience : estimation d'une trajectoire

Les données IMU collectées représentent les signaux d'accélération et de vitesse de rotation de l'appareil à tout instant. Afin d'en extraire une information qui a un sens physique de déplacement, elles sont passées aux fonctions d'estimation. Cette étape transforme ces sources en informations sur le déplacement que le piéton a effectué au cours d'un certain temps. i.e. un pas défini par son instant, sa longueur et son cap.

Le processus de validation de ces fonctions se constitue des tests unitaires présentés en annexe B. Chaque test suit un protocole qui dicte comment le téléphone est posé par rapport à l'utilisateur, l'action ou la trajectoire à effectuer par l'utilisateur ainsi que le résultat attendu. Alors que ce test vise à montrer le phénomène de dérive présent dans la PDR causée par l'intégration des données inertielles bruitées. La figure 4.7 montre que la trajectoire estimée fait environ 250 m. Le critère de retour au départ n'est pas valide dans cette réalisation. En terme d'erreur euclidienne, celle-ci est de 4.671 m en moyenne du au fait que la dérive angulaire est importante et que le tracé indique des trajectoires impossibles. i.e. traversant le mur.

L'erreur moyenne produite par la PDR est importante. Elle est reliée aux cumuls des erreurs sur l'estimation du cap et de la longueur de pas. Afin d'améliorer la localisation, il est nécessaire d'introduire une nouvelle source d'information au-delà des données inertielles. De la figure 4.7, on imagine ce que devient la trajectoire si les segments de la PDR pouvaient être recalés individuellement pour être cohérents avec la carte.

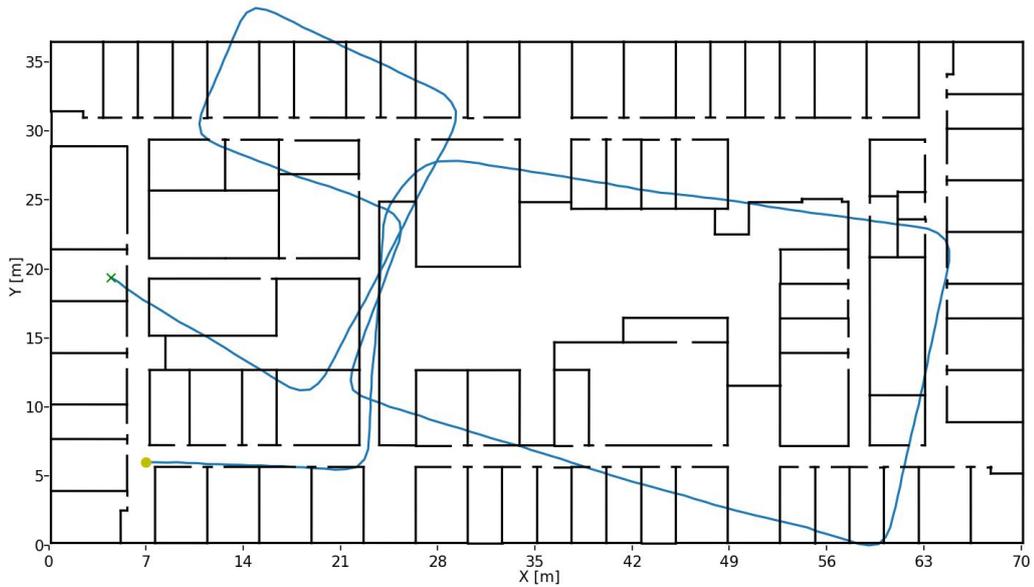


FIGURE 4.7 – Trajectoire estimée par PDR.

4.5 Récapitulatif

La navigation piétonne à l'estime (PDR) est une technique reposant sur le principe d'estimer récursivement la position en connaissant les position et cap de départ. Elle est composée de trois parties : la détection du pas, l'estimation de la longueur de pas SL et l'estimation de la différence du cap entre pas $d\theta$. La figure 4.8 résume la disposition actuelle des processus schématisés par des blocs.

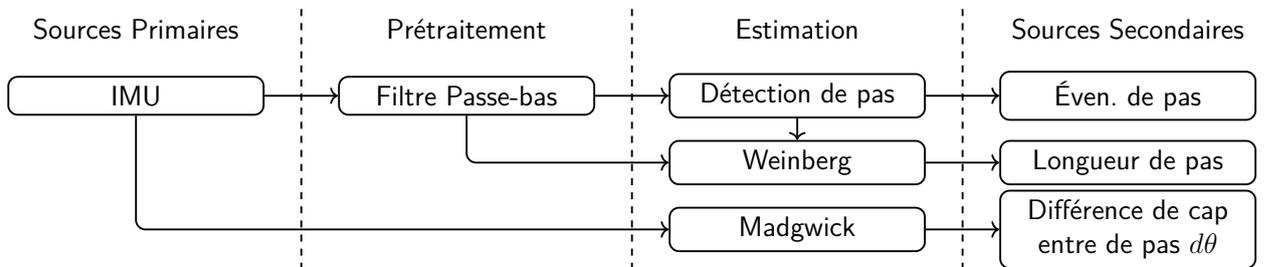


FIGURE 4.8 – Schéma bloc des fonctions présentées.

Avant de procéder à l'étape de détection des pas, il est nécessaire de filtrer le signal d'accélération avec un filtre passe bas dont la fréquence de coupure est de 2 Hz. Ensuite, la norme de ce signal est calculée puis centrée pour éliminer le vecteur de gravité. La détection du pas est ensuite effectuée par la technique de la détection des pics-vallées. Cette technique repose sur l'enchaînement d'événements dans le signal qui sont la transition des valeurs négatives vers positives, le maximum d'accélération correspondant au moment où le pied est dans la phase simple support, le passage des valeurs positives vers négatives, le minimum d'accélération qui correspond à la phase "*hill strike*" du modèle de marche et le passage des valeurs négatives vers positives qui finit le pas et signale le début du pas suivant.

Si cette séquence est respectée, l'événement de pas est émis. Les processus dépendant de cet événement dans la figure 4.8 calculent alors la longueur de pas SL et la différence de cap

$d\theta$. Plusieurs modèles d'estimation de SL sont présentées

Si ces étapes sont simples en apparence, les tests de validation (annexe B) conduits montrent que certains aspects du mouvement complexifient le déploiement de ces étapes. Le test de détection valide le fait que l'occurrence d'un pas est distinguable, périodique et discrétisable.

Nous avons noté que les modèles empiriques exprimaient le déplacement SL d'une manière très similaire de l'un à l'autre. L'équation déduisant la longueur du déplacement mêle une différence ou intégrale de l'accélération durant un pas et au moins un paramètre lié à l'utilisateur. La méthode de *Weinberg* semble avoir un bon compromis entre simplicité et adaptabilité.

L'autre problématique est l'estimation de cap où le cap du piéton dérive tout au long de la trajectoire à cause du bruit du gyroscope embarqué. Cet effet est négligeable sur des trajectoires courtes comme les tests l'ont montré, mais fausse totalement les trajectoires plus longues. Plusieurs méthodes ont été comparées dans l'état de l'art. Le filtre de *Madgwick* est l'un des estimateurs d'orientation le plus utilisé (≈ 2340 citations cumulées en 2022 pour [Madgwick et al., 2011]). Il permet de trouver les angles de roulis et de tangage dans le temps uniquement en exploitant les données d'accéléromètre et de gyroscope du fait que les données magnétiques soient rarement utilisables dans les espaces en intérieur. Le cap reste donc relatif à l'orientation de départ.

De ces deux problématiques, nous notons que les erreurs de calcul de la position et du cap à chaque pas s'accumulent. La trajectoire est vouée à dévier dans le temps, et l'erreur de positionnement est de plus en plus importante. Il y a donc la nécessité de corriger cette trajectoire via un signal qui ne soit pas proprioceptif.

Il existe une grande variété de systèmes de localisation en termes de modalité, précision désirée et application visée. Afin de juger des performances de ces systèmes, des métriques ont été définies dans l'état de l'art. Si on s'intéresse uniquement à celles permettant de juger des systèmes AAL, les métriques de distances euclidiennes et Mahalanobis en deux dimensions expriment la précision et l'incertitude de la localisation. La CDF permet de quantifier l'erreur tolérée que peut produire un système de localisation. Ces métriques rejoignent les critères choisis par le *framework EvAAL*.

Nous avons aussi abordé les approches pour construire la vérité terrain qui est nécessaire pour l'évaluation d'un système de localisation. Ces approches présentent toutes un compromis entre : résolution spatiale et temporelle, coût et temps de déploiement, intrusivité, synchronisation avec le système testé et capacité de mise à l'échelle. Nous avons opté pour la mise en place de points repères qui sont marqués au sol. Le piéton signale le passage sur ces points via l'application d'acquisition sur son téléphone. Cette méthode garantit une précision spatiale et temporelle satisfaisante comme montré dans le test associé à la vérité terrain.

Ce sont donc trois processus qui sont validés au bout de ce chapitre pour devenir des briques élémentaires utilisées dans la suite de notre étude. Ces briques sont essentielles pour pouvoir accomplir les objectifs [A, B & C].

4.6 Conclusion

Dans ce chapitre, nous avons présenté la navigation piétonne à l'estime (PDR). Ensuite, les trois étapes de celle-ci que sont : la détection du pas, l'estimation de la longueur de pas SL et l'estimation de la différence du cap entre pas $d\theta$, sont présentées. Les modèles d'estimation de SL sont exposés. Les métriques d'évaluation pour les systèmes de localisation tirées du

framework EvAAL sont présentées. Une section aborde les méthodes pour établir une vérité terrain pour les expériences testant un système de localisation. Enfin, une série de tests de validation est présentée afin de valider les concepts exposés et pour les utiliser dans notre prochaine étape : la correction de trajectoire par sources secondaires.

Chapitre 5

Correction de la trajectoire par fusion de sources secondaires

Sommaire

5.1	Fusion par filtre récursif	82
5.1.1	Filtrage bayésien	82
5.1.2	Modèle d'état dans le filtre particulaire PF	84
5.2	Étapes du filtrage à particules	85
5.2.1	Initialisation	85
5.2.2	Prédiction	87
5.2.3	Mise à jour	89
5.2.4	Ré-échantillonnage	90
5.3	Filtrage par murs	92
5.3.1	Formats de carte	92
5.3.2	Contraintes géométriques pour filtre particulaire	95
5.3.3	Régénération	95
5.3.4	Sortie du filtre	96
5.3.5	Sortie du filtre contrainte	97
5.4	Filtrage par modèle de mouvement humain	97
5.4.1	Grille de vraisemblance	98
5.4.2	Fonctions d'appartenance	98
5.5	Expériences	99
5.5.1	Étude de l'effet du nombre de particules	100
5.5.2	Étude de l'effet du gain de Weinberg	101
5.5.3	Trajectoire réalisée	102
5.5.4	Effet de la grille de vraisemblance	104
5.6	Récapitulatif	108
5.7	Conclusion	110

Dans le chapitre précédent, nous avons noté que les données inertielles comprenaient certaines erreurs dans les mesures acquises. Nous avons aussi constaté que les méthodes d'estimation des sources secondaires (longueur de pas et différence de cap) à partir de ces sources primaires (signaux d'accéléromètres et de gyroscopes) n'étaient pas suffisantes à elles seules pour corriger ces erreurs. Il est donc nécessaire d'ajouter des processus en aval de ces sources secondaires pour affiner l'estimation de la position à chaque pas. Pour cela, on procède à la fusion de données par un filtre récursif mentionné dans la section 2.5.1.4. Plus particulièrement, le filtre à particules (PF) est l'objet de notre étude dans ce chapitre. Son principe de fonctionnement ainsi que toutes ses étapes sont présentées. Nos contributions et les choix d'instanciation sont aussi présentés. Le système présenté dans ce chapitre est récapitulé par la figure 5.1 qui est une évolution de la figure 4.8.

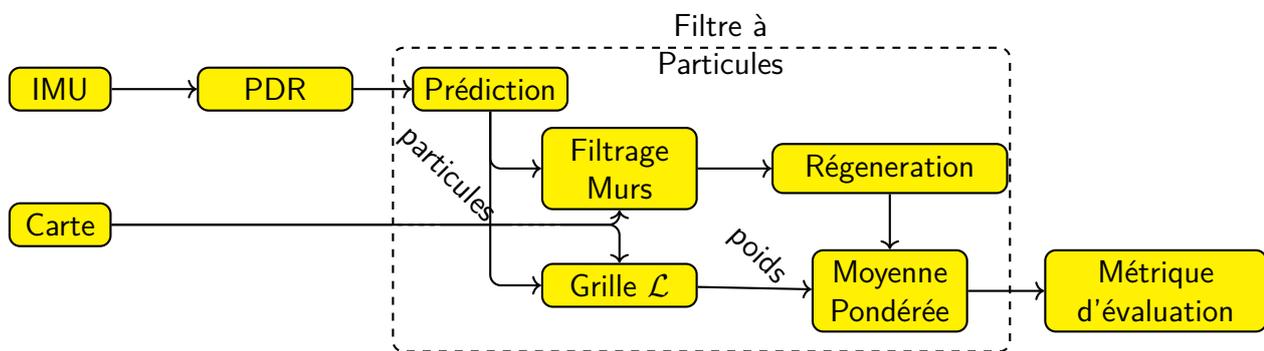


FIGURE 5.1 – Schéma récapitulatif du système actuel pour un pas.

5.1 Fusion par filtre récursif

5.1.1 Filtrage bayésien

On souhaite estimer la position et le cap successifs qu'un piéton prend à chaque pas. Il s'agit de résoudre un problème d'estimation récursive d'un état qui représente un piéton. En se basant sur les définitions présentées en section 3.2, le piéton est représenté par l'état E tel que :

$$E_t = \{[x_t, y_t]^T, \theta_t\} \quad (5.1)$$

Les opérations que subit cet état dans ce chapitre sont linéaires et donc il peut être réécrit sous la forme d'un vecteur $E_t \in \mathbb{R}^3$ même si les composantes ne sont pas homogènes. Ce vecteur d'état représente le piéton à une itération t :

$$E_t = [x_t, y_t, \theta_t]^T \quad (5.2)$$

Selon [Gordon et al., 1993], l'équation de transition d'état est :

$$E_t = f(E_{t-1}, \omega_t) \quad (5.3)$$

Avec $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, la fonction de transition du système et $\omega_t \in \mathbb{R}^m$ le terme de bruit dont la distribution est indépendante du temps [Carpenter et al., 1999]. Une observation, notée $D_t \in \mathbb{R}^p$, est formalisée à chaque itération t .

$$D_t = h_t(E_t, \nu_t) \quad (5.4)$$

Avec $h_t : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$, la fonction de mesure et $\nu_t \in \mathbb{R}^r$, un terme de bruit dont la distribution est indépendante du système et du temps.

Le but est de construire la fonction de Densité de Probabilité (PDF) de l'état actuel E_t : $p(E_t|D_t)$ en utilisant toute information D_t disponible à cette itération là. On calcule cette PDF via un processus récursif à deux étapes : prédiction et mise à jour.

Si on suppose $p(E_{t-1}|D_{t-1})$ la PDF à l'itération $t - 1$ connue, il est possible d'obtenir la PDF *a priori* à l'itération t :

$$p(E_t|D_{t-1}) = \int p(E_t|E_{t-1})p(E_{t-1}|D_{t-1})dE_{t-1} \quad (5.5)$$

Où $p(E_t|E_{t-1})$ est un modèle de Markov de l'évolution d'état défini par l'équation du système et les propriétés statistiques connues de ω_{t-1} [Carpenter et al., 1999].

$$p(E_t|E_{t-1}) = \int p(E_t|E_{t-1}, \omega_{t-1})p(\omega_{t-1}|E_{t-1})d\omega_{t-1} \quad (5.6)$$

Vu l'hypothèse sur l'indépendance du bruit de processus, $p(\omega_{t-1}|E_{t-1}) = p(\omega_{t-1})$, on a :

$$p(E_t|E_{t-1}) = \int \delta(E_t - f(E_{t-1}, \omega_{t-1}))p(\omega_{t-1})d\omega_{t-1} \quad (5.7)$$

Avec $\delta(\cdot)$ la fonction de *Dirac* qui exprime le fait que si les états et bruits précédents sont connus, alors l'état actuel peut être calculé via l'équation (5.3) [Gordon et al., 1993]. Dès qu'une mesure notée D_t est disponible, une mise à jour de la PDF *a posteriori* est faite via la règle de *Bayes* :

$$p(E_t|D_t) = \frac{p(D_t|E_t)p(E_t|D_{t-1})}{p(D_t|D_{t-1})} \quad (5.8)$$

Où le dénominateur de normalisation est :

$$p(D_t|D_{t-1}) = \int p(D_t|E_t)p(E_t|D_{t-1})dE_t \quad (5.9)$$

Dans un deuxième temps, la PDF conditionnelle $p(D_t|E_t)$ est définie par le modèle de mesure et les propriétés statistiques de ν_t qui, de façon similaire à ω_t , est aussi indépendant de l'état précédent.

$$p(D_t|E_t) = \int \delta(D_t - h_t(E_t, \nu_t))p(\nu_t)d\nu_t \quad (5.10)$$

Dans l'équation (5.8), la mesure D_t met à jour l'état prédit à partir de l'instant précédent pour obtenir l'état *a posteriori*.

La solution formelle au problème d'estimation Bayésienne récursive est représentée par la relation de récurrence des équations (5.5) et (5.8). Il est admis que dans la plupart des cas ces équations n'ont pas de solutions sans introduire plus d'hypothèses et de restrictions sur les modèles de système et de mesure [Jouin et al., 2016]. Une méthode pour résoudre ces équations est le filtre particulaire (PF).

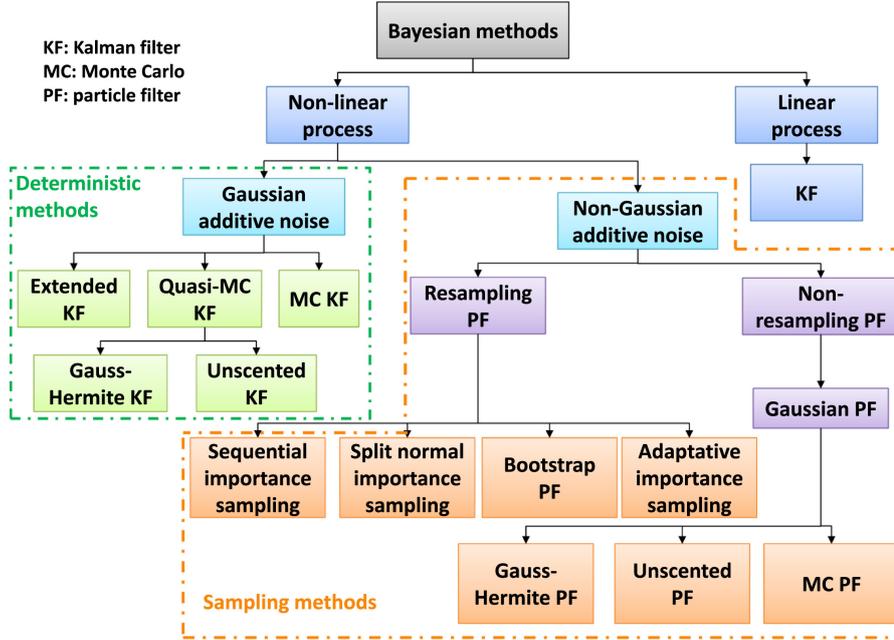


FIGURE 5.2 – Schéma de positionnement du filtrage à particules de [Jouin et al., 2016].

La figure 5.2 montre qu’il existe plusieurs variantes du PF [Jouin et al., 2016, Sikorska et al., 2011]. C’est pour cela que par la suite, les détails de l’instanciation du PF dans le cadre de notre problème de localisation sont présentés.

5.1.2 Modèle d’état dans le filtre particulaire PF

Le PF [Gordon et al., 1993] est un outil d’estimation d’état *a posteriori* non linéaire non gaussien. Il résout le problème d’estimation Bayésienne récurrente en approximant la PDF *a posteriori* par un ensemble fini de N particules. La distribution de ces particules dans l’univers des variables aléatoires est appelé l’*espace des possibles*. Le filtre vise à modéliser cet espace et suivre son évolution au cours du temps/événements avec l’ensemble des N particules. Il considère que la physique d’un problème peut être modélisée par un processus stochastique.

Dans la localisation piétonne, le PF exploite les sources d’informations secondaires issues d’informations fournies par méthode d’estimation de la navigation piétonne en émettant N particules avec chacune un bruit de propagation sur la longueur et le cap de pas. Ces particules représentent des hypothèses sur les trajectoires possibles que le piéton a pu prendre pendant le pas. Chaque particule reçoit la même information de prédiction mais a un tirage de variable aléatoire indépendant des autres particules (aucune intradépendance) et des pas précédents (aucune interdépendance). L’historique d’une particule constitue une trajectoire possible.

On modélise donc une particule par un vecteur d’état composé de la position et l’orientation du piéton en accord avec nos hypothèses de la section 3.3 tel que :

$$E_t = [x_t, \quad y_t, \quad \theta_t]^T \quad | \quad E_t \in \mathbb{R}^3 \quad (5.11)$$

Et un vecteur de prédiction U_t décrivant les déplacements sur x et y et la différence cap du pas $d\theta$:

$$U_t = [dx_t, \quad dy_t, \quad d\theta_t]^T \quad | \quad U_t \in \mathbb{R}^3 \quad (5.12)$$

Avec $dx_t = SL_t \times \cos \theta_t$ et $dy_t = SL_t \times \sin \theta_t$ de la section 4.1. Enfin, le vecteur de bruits de processus est ajouté :

$$E_t = E_{t-1} + U_t + N_t \quad (5.13)$$

$$\text{avec } N_t = [\delta_t \cos(\theta_t), \quad \delta_t \sin(\theta_t), \quad \alpha_t]^T \quad (5.14)$$

Où δ et α sont les variables aléatoires représentant les bruits additifs sur la longueur du pas SL et $d\theta$ respectivement tirés de des distributions gaussiennes présentées dans le tableau 5.1.

Le filtre associe alors à chaque particule i , à l'itération t , un vecteur d'état E_t^i tel que :

$$E_t^i = E_{t-1}^i + U_t^i + N_t^i \quad (5.15)$$

En remplaçant les équations (5.11), (5.12) et (5.14) dans l'équation (5.15), on obtient :

$$E_t^i = \begin{bmatrix} x_t^i \\ y_t^i \\ \theta_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i \\ y_{t-1}^i \\ \theta_{t-1}^i \end{bmatrix} + \begin{bmatrix} SL_t^i \times \cos(\theta_t^i) \\ SL_t^i \times \sin(\theta_t^i) \\ d\theta_t^i \end{bmatrix} + \begin{bmatrix} \delta_t^i \times \cos(\theta_t^i) \\ \delta_t^i \times \sin(\theta_t^i) \\ \alpha_t^i \end{bmatrix} \quad (5.16)$$

Avec $x_0 = X_0$, $y_0 = Y_0$, $\theta_0 = \Theta_0$ comme conditions initiales.

Réduit à l'équation (5.16), le PF représente de manière égale toutes ses particules et donc pave uniformément l'espaces des hypothèses. On dit que le filtre n'est pas probabiliste, mais qu'il fonctionne seulement en mode multi-hypothèses. Il permet de pister et suivre l'évolution de l'espace des hypothèses mais ne comprend aucune notion d'importance parmi les différentes informations qu'il contient. Il est donc nécessaire d'exprimer cette importance par un facteur : le poids d'une particule w_t^i . Dans ce cas, une particule E_t^i est représentée par :

$$E_t^i = \begin{cases} \begin{bmatrix} x_t^i \\ y_t^i \\ \theta_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i \\ y_{t-1}^i \\ \theta_{t-1}^i \end{bmatrix} + \begin{bmatrix} SL_t^i \times \cos(\theta_t^i) \\ SL_t^i \times \sin(\theta_t^i) \\ d\theta_t^i \end{bmatrix} + \begin{bmatrix} \delta_t^i \times \cos(\theta_t^i) \\ \delta_t^i \times \sin(\theta_t^i) \\ \alpha_t^i \end{bmatrix} \\ w_t^i = w_{t-1}^i \times \mathcal{L}(\tilde{E}_t^i) \end{cases} \quad (5.17)$$

Où $\mathcal{L}(\tilde{E}_t^i)$ est la vraisemblance liée à l'information externe issue de l'environnement dans lequel la i ème particule est après l'étape de prédiction. Cette vraisemblance est présentée avec plus de détails dans la section 5.4.

5.2 Étapes du filtrage à particules

Le filtre procède par 4 étapes essentielles [Jouin et al., 2016] qui sont : l'initialisation, la prédiction, la mise à jour et le ré-échantillonnage. Dans la partie qui suit, chacune de ces étapes est expliquée en illustrant différentes configurations et paramètres de celles-ci ainsi que les modifications que nous y avons introduites.

5.2.1 Initialisation

À l'instant $t = 0$, le PF dispose de N particules afin de représenter l'espace des hypothèses de départ. Dans l'état de l'art de la localisation, l'initialisation est soit globalisée [Seigneur,

2006, Potortì, 2020, Renaudin, 2019, Karkus et al., 2018] ou localisée [Potortì, 2020, Renaudin, 2019, Lu et al., 2019, Ning and Chen, 2019, Zhao et al., 2019, Yu et al., 2017]. Chacune des deux approches possède des avantages et des inconvénients et des *a priori* différents.

5.2.1.1 Globalisée

Dans ce cas, les N_0 particules initiales sont distribuées uniformément sur l'univers des possibles. Le PF ne connaît *a priori* pas la position ni le cap de départ du piéton. Le but de cette initialisation est de permettre aux PF de couvrir l'univers des possibles avec assez de particules et de progressivement éliminer celles qui sont mensongères. Le PF converge vers un nombre de particules, $N < N_0$, représentant le nombre d'échantillons nécessaires pour qu'il puisse modéliser l'espace des hypothèses et suivre son évolution en fonction du temps en fonctionnement nominal. La figure 5.3 montre les positions des particules à l'instant $t=0$. i.e. avant même les premières prédictions et observations.

Cette méthode a pour avantage de répondre à la problématique dite du "robot perdu" de manière assez simple. Un premier défaut est qu'il lui faut un nombre inconnu d'itérations avant de converger sans avoir la garantie de convergence. Mais, son défaut majeur reste le nombre de particules très important à l'initialisation : au moins 100 fois le nombre nominal N . Il se peut aussi que cette méthode exige un historique des particules avant la convergence. Ceci fait que beaucoup de systèmes n'ont ni la puissance de calcul et ni l'espace en mémoire requis pour pouvoir embarquer cet algorithme d'initialisation.

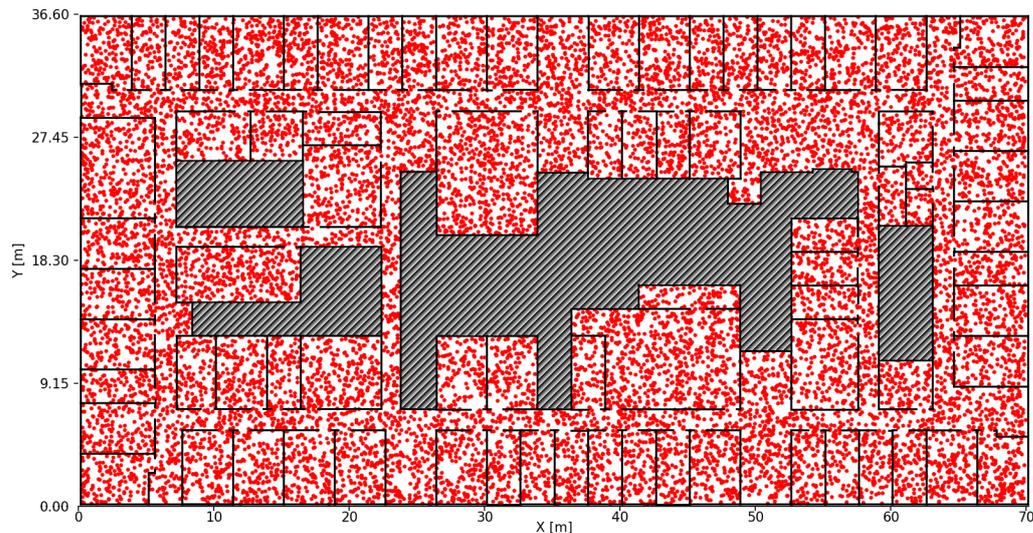


FIGURE 5.3 – Initialisation globale sur la carte avec 20000 particules.

5.2.1.2 Localisée

L'*a priori* ici est que la position de départ, (x_0, y_0) , du piéton est exactement connue ou au moins bornée dans l'environnement intérieur. Dans ce cas, le PF peut échantillonner aux alentours de (x_0, y_0) les N particules nécessaires à son fonctionnement en ajoutant un bruit sur les variables d'état x, y . Si le cap de départ Θ_0 est lui aussi connu, il peut générer les particules afin qu'elles soient orientées vers ce cap tout en ajoutant un bruit d'initialisation. Un exemple de cette initialisation localisée avec position et cap de départ connu, est montré

en figure 5.4. Les bruits d'initialisation sont $\delta_0 \sim \mathcal{N}(0, 0.05)$ en [m] et $\alpha_0 \sim \mathcal{N}(0, \pi/10)$. en [rad]. La position de départ est (0,0). le cap de départ est 0 rad.

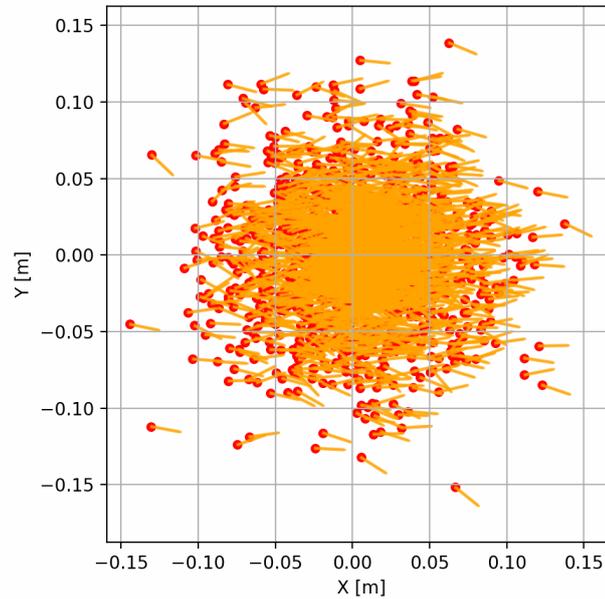


FIGURE 5.4 – Initialisation de 2000 particules avec les bruits δ_0 et α_0 .

Cette approche d'initialisation a pour mérite de ne pas être aussi coûteuse en ressources que sa variante globale. Il n'y a aussi aucun critère sur la convergence car le système commence dès l'instant $t = 0$ à localiser le piéton. Le coût en mémoire est prévisible et constant avec N particules. L'un des arguments contre cette méthode est qu'il existe un *a priori* relativement important sur l'état initial qui n'est pas toujours disponible ou correct. Mais, en réalité, dans les espaces intérieurs le nombre d'entrées dans un bâtiment est fortement limité (bâtiments Digitéo, ENS, CentraleSupélec : 1, 2, 2 entrées sans badge, une vingtaine de sorties chacun). Donc, l'utilisateur vient souvent d'un endroit localisé grâce à des balises de type tourniquet ou similaire, ou un positionnement GNSS en extérieur.

5.2.2 Prédiction

L'étape de prédiction consiste à utiliser le vecteur de sources secondaires produits par les processus présentés en section 4.1 afin de modifier l'état des particules pour prédire l'état *a posteriori*. Dans le cas de notre localisation, vu que chaque particule représente une hypothèse, donc une trajectoire possible, on peut prédire le déplacement du pas effectué avec sa longueur SL_t et sa variation de cap $d\theta_t$. En, ajoutant les bruits : δ en [m] sur la longueur de pas et α en [°] sur le cap à chaque prédiction (à chaque pas), le PF permet de modéliser l'espace des hypothèses :

- en corrigeant la dérive angulaire vue en section 3.1.3 qui est liée à l'objectif [B].
- en introduisant le bruit de l'accéléromètre vu en section 3.1.3.
- en introduisant un bruit modélisant l'erreur d'estimation de la longueur de pas par la méthode de *Weinberg* vue en section 4.1.3.1 qui est liée à l'objectif [A].
- en introduisant un bruit modélisant l'erreur d'estimation du cap par la filtre de *Madgwick* vue en section 4.1.4.

— en compensant l'angle de tangage et la dérive sur l'axe vertical vus en section B.1.8

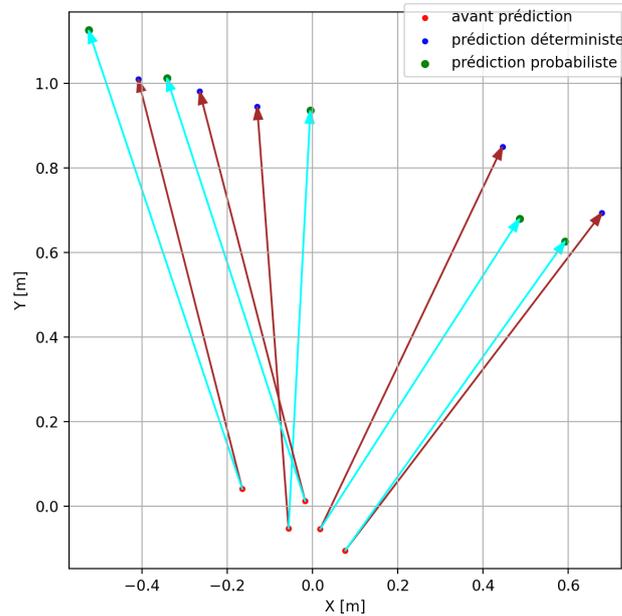


FIGURE 5.5 – Comparaison entre prédictions déterministe et probabiliste pour 5 particules.

Un exemple de prédiction est donné dans la figure 5.5. Elle montre à la fois une prédiction dite déterministe où les bruits de propagation α et δ sont nuls et le cas où ils sont non nuls. Dans le premier cas, chaque particule a la même distance prédite (segments marrons tous de longueur égale). Quand on compare celle-ci à la prédiction probabiliste avec des bruits de propagation non-nuls, on a une différence dans le cap et la distance prédits.

Les bruits de propagation ont donc un effet important sur la modélisation de la physique du problème par le PF. Les distributions d'où sont tirés α et δ ont un impact sur cette modélisation. Dans la figure 5.6, Le PF produit la prédiction d'un million de particules avec les paramètres fournis dans le tableau 5.1 en moyenne et écart-type. Les prédictions résultantes en bleu forment la combinaisons de 2 gaussiennes : la première est en forme de cône d'ouverture sur le cap. La deuxième est un segment de couronne [Varlamov and Aslamazov, 2012]¹ i.e. un disque borné par 2 rayons et 2 angles. À titre d'exemple, d'autres tracés ont été réalisés pour différentes distributions de bruits de propagation avec des bruits d'initialisation nuls sur la figure 5.6.

variable d'état	bruit	initial	de prédiction	U_t
x, y [(m,m)]	δ	$\mathcal{N} \sim (0, 0.025)$	$\mathcal{N} \sim (0, 0.15)$	$SL_t = 0.75$
θ [rad]	α	$\mathcal{N} \sim (0, \pi/80)$	$\mathcal{N} \sim (0, \pi/45)$	$d\theta_t = \pi/2$

TABLE 5.1 – Paramètres du PF pour la figure 5.6b.

1. Spherical Shell - Wolfram - <https://mathworld.wolfram.com/SphericalShell.html>

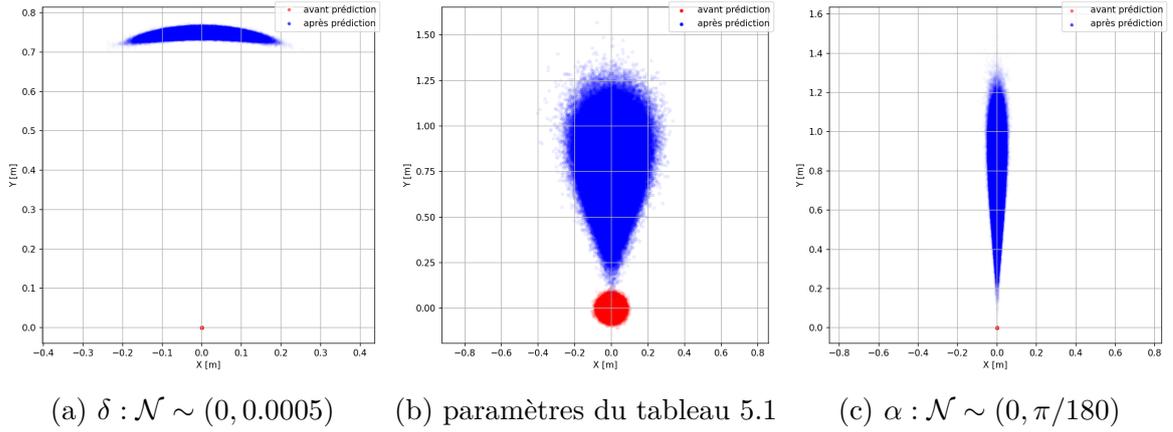


FIGURE 5.6 – Prédictions du déplacement avec 10^6 particules.

5.2.3 Mise à jour

Une fois la prédiction pour chaque particule faite et leur vecteur d'état prédit, noté \tilde{P}_t^i , le PF doit évaluer à quel point ces prédictions sont vraisemblables. Afin de permettre une modélisation correcte de l'espace des hypothèses, chaque particule doit recevoir un coefficient qui indique l'importance de l'information qu'elle contient. Ce coefficient est appelé un poids et est noté w_t^i . Il est associé à la vraisemblance qu'une particule prédise un vecteur d'état sachant une distribution prédéfinie sur les prédictions. Cette vraisemblance \mathcal{L} n'est pas toujours évidente à définir et dépend fortement du contexte applicatif du PF. Dans le cas où cette vraisemblance est uniforme sur tout l'espace des hypothèses, comme dans [Lu et al., 2019, Zhang et al., 2019], le PF passe en mode multi-hypothèses du fait que toutes les particules ont le même poids $1/N$ et donc la même importance. Nous présentons dans la section 5.4 notre approche de pondération des particules publiée dans [Ghaoui et al., 2022].

Après mise à jour, les poids des particules sont normalisés afin que l'équation (5.18) soit vérifiée :

$$\sum_{i=1}^N w_t^i = 1 \quad (5.18)$$

L'étape de mise à jour permet donc de déformer l'espace des possibles en fonction de l'information disponible après l'étape de prédiction. Cette information peut être contextuelle et dépendante de l'environnement dans lequel une particule se situe au moment de cette mise à jour. Mais, la mise à jour peut engendrer le phénomène de dégénérescence.

5.2.3.1 Dégénérescence

Au fur des itérations successives, les particules moins vraisemblables vont progressivement perdre de leur importance jusqu'à avoir un poids insignifiant. Dans ce cas, ces particules et leurs informations ne sont plus des hypothèses vraisemblables. Mais, elles occupent toujours des échantillons parmi les N totaux. Et en même temps, une particule va gagner progressivement en importance jusqu'à devenir la seule information représentative avec un poids très proche de 1. Ceci limite l'expressivité du PF selon [Elfring et al., 2021]. Une visualisation de ce

phénomène est proposée dans la figure 5.7 où certaines particules détiennent une grande partie de l’information alors que beaucoup de particules ont un poids très faible.

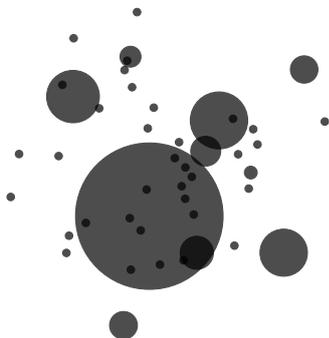


FIGURE 5.7 – Phénomène de dégénérescence visualisé.

Ce qui fait que progressivement le PF a un nombre effectif de particules réduit de $N_{eff} < N$. On parle alors de dégénérescence du PF car il échoue à représenter correctement l’espace des possibles malgré l’usage de N particules. Ce phénomène est une fatalité dans tout PF [Elfring et al., 2021]. Il doit donc être détecté et traité par un processus : le ré-échantillonnage.

5.2.4 Ré-échantillonnage

5.2.4.1 Définition

C’est le processus qui, en partant d’un vecteur \mathbf{w} de N poids de particules, produit un vecteur \mathbf{o} de N indices. Le processus duplique les particules aux poids les plus forts et éliminent les particules aux poids les plus faibles. Plus formellement, le ré-échantillonnage est défini comme l’application suivante :

$$f : \mathbf{w} \in [0, 1]^N \longrightarrow \mathbf{o} \in \mathbb{N}^N \quad (5.19)$$

Selon [Murray et al., 2016], il existe plusieurs algorithmes acceptables pour décrire ce ré-échantillonnage. Du moment où le résultat est aléatoire, la probabilité d’avoir d’un indice o_t^i sachant l’ensemble de poids précédents, \mathbf{w}_{t-1} , peut être exprimée comme :

$$\mathbb{E} \left(o_t^i \mid \mathbf{w}_{t-1} \right) = \frac{N w_{t-1}^i}{\sum_{j=1}^N w_{t-1}^j} \quad (5.20)$$

i.e. que le nombre de descendants d’une particule est égale à son poids normalisé multiplié par N .

Le ré-échantillonnage est considéré comme une étape essentielle dans le PF [Gordon et al., 1993, Bolic et al., 2005] car il permet de contrer la dégénérescence du PF. Ceci permet au PF de continuer à représenter correctement l’espace des hypothèses. Ce processus a bien été étudié [Elvira et al., 2017, Hol et al., 2006, Bolic et al., 2005, Murray et al., 2016, Douc and Cappe, 2005] et il existe plusieurs implantations pour filtre particulaire. Néanmoins, ces implantations se regroupent principalement en 6 groupes tirés de [Douc and Cappe, 2005, Murray et al., 2016] qui sont mentionnées par la suite.

Chacun de ces algorithmes a des particularités et propriétés sur le résultat de sortie, la variance des duplications, le biais de ré-échantillonnage, la stabilité, la complexité algorithmique, l’empreinte mémoire, l’intensité arithmétique et la parallélisabilité. L’étude de chacun de ces aspects n’est pas abordée dans ce manuscrit. Quelque soit l’algorithme choisi, il sera le résultat d’un compromis entre tous ces facteurs et devra être adapté au problème traité. En d’autres termes, il n’y a pas de choix universel.

Différentes familles d’algorithmes de ré-échantillonnage et leurs instanciations sont présentées dans l’annexe B.4. Parmi ces algorithmes de ré-échantillonnage, nous avons opté pour l’implantation d’une variante de l’algorithme multinomial sous la forme d’une régénération par sous ensemble présentée en section 5.3.3 afin de correspondre à notre cas.

5.2.4.2 Appauvrissement des échantillons

La résolution du problème de dégénérescence de la section 5.2.3.1 mène à un autre cas problématique. Pendant l’étape prédiction, si le bruit du processus stochastique est trop faible, les particules ne sont pas suffisamment dispersées pour représenter correctement l’espace des hypothèses. Le ré-échantillonnage accentue ce phénomène.

Il y a effectivement un amas de particules condensées aux poids très importants qui représente le même état ou des états fortement proches. Cet amas de particules, après quelques itérations, se concentre sur un seul point [Elfring et al., 2021]. L’effet de cette concentration est comparable à celui d’un trou noir. Il déforme l’espace des hypothèses et le concentre en un seul point, appauvrissant le filtre, d’où le nom donné à ce phénomène : *Appauvrissement des échantillons*.

Il est possible de palier ce problème en mettant des bruits de propagation plus importants. Cette solution n’est pas considérée acceptable car ces bruits ont un sens physique et sont là pour représenter la modélisation du problème traité. L’autre solution est d’éviter de ré-échantillonner trop fréquemment afin de laisser au PF le temps de propager ses particules dans l’espace des hypothèses. Il faut donc définir une condition pour appeler le processus de ré-échantillonnage.

Critères de déclenchement du processus de ré-échantillonnage est défini comme un prédicat qui a pour entrée le vecteur des poids \mathbf{w}_t ou les particules. Il doit indiquer au PF si un ré-échantillonnage est nécessaire en calculant une grandeur statistique sur la distribution en entrée. Différents critères existent pour accomplir cela. Parmi ceux-ci, on présente celui de [Kong et al., 1994] et de [Pham et al., 2003].

Kong Ce critère vise à calculer N_{eff} le nombre efficace de particules tel que :

$$N_{eff} \approx \frac{1}{\sum_{i=1}^N (w^i)^2} \quad (5.21)$$

Alors, il suffit d’établir un seuil N_{th} tel que $N_{eff} < N_{th}$. Ce seuil est déterminé empiriquement.

Pham Un autre critère est celui de l’entropie définie par l’équation (5.22). La valeur de l’entropie permet de qualifier rapidement si le filtre est appauvri ou a dégénéré.

$$S = \log N + \sum_{i=1}^N w^i \log(w^i) \quad (5.22)$$

L'équation (5.23) représente le cas où toutes les particules ont le même poids. Alors que l'équation (5.24) représente le cas où il existe un poids proche de 1.

$$Si : S \approx 0 \iff w^i \approx \frac{1}{N} \quad | \quad \forall i \in \{1, \dots, N\} \quad (5.23)$$

$$Si : S \approx N \iff w^j \approx 1 \quad | \quad \exists j \neq i \in \{1, \dots, N\} \quad (5.24)$$

Un seuil S_{th} doit être fixé tel que $S > S_{th}$ comme pour le critère précédent.

5.3 Filtrage par murs

Dans le cas de la localisation en intérieur, une information est la présence de contraintes géométriques permettant de définir un ensemble de règles qui régissent l'évolution du mouvement humain. Les cartes sont quasi-systématiquement disponibles sous différents formats.

5.3.1 Formats de carte

En matière de cartes, peu d'ouvrages exploitent des plans préexistants de bâtiments. Selon [Chen and Clarke, 2020], les cartes se présentent sous différents types, ont des informations intégrées, différents niveaux de détail et précisions de mesures, les éléments sémantiques inclus, les relations topologiques ou les textures et matériaux des éléments

5.3.1.1 Structures Raster ou Vecteur

Le raster, venant de l'infographie, est un format qui consiste en un ensemble de lignes de pixels. L'objet numérique est une image de forme matricielle, souvent sous la forme de couches superposées. La superposition de ces couches permet de représenter différents types d'informations sur une même zone géographique.

Au contraire le format Vecteur encode la géométrie des éléments le composant (cercle, carré,...) pour pouvoir produire une image adaptable en fonction de la tâche que la carte doit réalisée (être affichée, imprimée,...). Ce format est plus avancé que le Raster et a généralement une empreinte mémoire plus élevée.

Ces deux formats constituent souvent la base pour ceux qui suivent. En localisation, le raster sert très souvent à seulement visualiser les trajectoires estimées afin de donner au lecteur une idée plus claire sur le mouvement effectué. Les vecteurs quant à eux peuvent fournir plus efficacement les informations géométriques pour les convertir en informations géographiques. Dans les deux cas, l'étape de collection de la carte et de conversion du format est inévitable pour les adapter au problème. Donc, ces formats sont considérés comme une source primaire de données.

5.3.1.2 Plan d'étage

C'est le format le plus commun et directement disponible dans un bâtiment. Il consiste en une image 2D de la vue de haut de l'étage. Il y représente différents artefacts cartographiques tels que les murs, sorties, équipements contre incendies et/ou identification des lieux. Les murs peuvent être représentés comme une collection de segments de 2 points chacun. Les éléments du plan sont localisés par rapport à l'origine du plan (souvent (0,0) = sud ouest).

5.3.1.3 *Point Cloud Data* (PCD)

Ce format est une collection de coordonnées 3D dans \mathbb{R}^3 encodant les positions de points sous la forme de nuage (d'où le nom). Ce nuage est produit par un capteur LiDAR qui consiste en plusieurs lasers que l'on appelle "nappe". Une nappe émet un rayon à un angle et calcule le temps d'aller-retour (ou non retour) du faisceau pour évaluer la distance à l'obstacle. Les LiDARs modernes possèdent facilement 64 nappes où chaque nappe a un angle de tangage plus faible que la précédente. Par exemple, le LiDAR *Velodyne HDL-32E²* collecte jusqu'à 1.39 million d'échantillons par seconde avec une précision de ± 2 cm sur un angle d'ouverture au tangage de $[10, -30]^\circ$ sur 360° d'azimut.

Malgré la précision, la richesse et la densité d'informations de ce format, il n'est pas adapté comme support de localisation à lui seul. Ceci est dû au fait que les données des points n'encodent aucune relation géographique ni topologique entre elles. Enfin, il requiert un débit conséquent dû au volume important des données incluses, pour 3 flottants 32 bits : 1.39×12 Mo/s ce qui n'est pas en accord avec les objectifs [F & G].

5.3.1.4 *Geographic Information System* (GIS)

Le GIS est une base de donnée contenant des données géographiques. Il permet de collecter, stocker, traiter, mettre à jour, analyser et présenter tout type de données géographiques ou spatiales. Plusieurs applications de localisation se basent sur un GIS car il permet de réunir des informations hétérogènes : coordonnées, temps, adresses, relations spatiales et logiques et de les représenter autour d'un même référentiel.

5.3.1.5 *Building Information Model* (BIM)

Le BIM est le modèle présentant une modélisation très fine de l'espace intérieur et un niveau de détail des plus élevés, jusqu'au matériaux utilisés pour la construction. Il sert à assister des tâches où la précision demandée est extrêmement élevée [Chen and Clarke, 2020], par exemple : la modélisation de la propagation de chaleur dans un espace intérieur ; la simulation de propagation de virus [Armand and Tâche, 2022] ; la localisation et navigation dans des espaces 3D en réalité physique, virtuelle ou mixte [Guerra et al., 2019].

5.3.1.6 Graphe

Le graphe est l'outil adapté pour exprimer les liens logiques ou topologiques entre les différents éléments dans un espace en intérieur. En définissant un arc comme une transition et un nœud comme une entité d'intérêt, il est possible de se localiser au nœud près.

2. <https://velodynelidar.com/products/hdl-32e/>

Un paramètre important lors de la construction d'un graphe à partir d'une source primaire de cartographie est le choix du pas d'échantillonnage spatial. Un nœud peut représenter : un étage, une zone, une salle, une chaise,... respectivement l'arc peut représenter : un ascenseur ou escalier, des halles de transition, les portes, l'espace entre meubles,... Les graphes sont donc une représentation efficace de l'espace.

Toutefois, dans les espaces ouverts, l'efficacité des graphes diminue. De plus, en raison du pas d'échantillonnage spatial, l'empreinte mémoire d'un graphe n'est pas prévisible. En outre, le temps d'accès à un élément dans un graphe est soumis à une contrainte de navigation de graphe, c'est-à-dire passer d'un nœud à un autre. Dans la figure 5.8, une salle est représentée par au moins 2 nœuds, et chaque transition entre salle et couloir nécessite 2 autres nœuds.

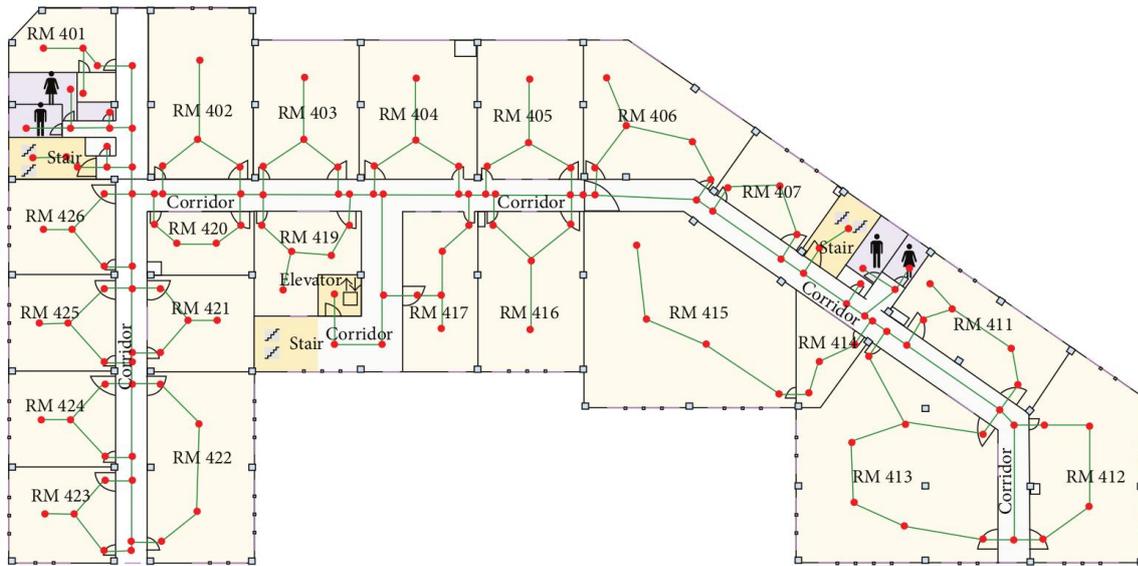


FIGURE 5.8 – Représentation d'un espace en intérieur par un graphe. Image de [Shang et al., 2015].

5.3.1.7 Grille

En localisation, la grille est souvent utilisée comme une grille d'occupation [Saarinen et al., 2012, Dang et al., 2015, Meyer-Delius et al., 2012] qui vise à représenter l'environnement par une variable aléatoire de champ régulièrement espacée et à indiquer la probabilité qu'une case soit occupée à un instant t . Il existe d'autres manières d'utiliser les grilles en localisation notamment celle que l'on présente dans la section 5.4.

5.3.1.8 Tracé à la main

Malgré la disponibilité des cartes quelque soient leur forme, il existe des cas où une carte peut être manquante, imprécise, fautive, dépassée ou inexploitable pour une raison quelconque. Dans ce cas, il reste quand même possible d'utiliser une approximation tracée par l'utilisateur à main levée. Par exemple, [Luperto et al., 2020] exploite une carte incomplète comme *a priori* pour aider à la localisation de son robot. Le tracé à main levée est raffiné et mis à jour par la cartographie faite par le robot.

5.3.2 Contraintes géométriques pour filtre particulaire

Une des conclusions de la section 2.5.1.4 est que le PF est très bien adapté à l'application de contraintes extraites d'information géométriques de la carte. Car, il représente la PDF *a posteriori* via l'ensemble des particules discrètes. Une hypothèse sur le déplacement est que le piéton ne traverse jamais les murs dans un bâtiment et ne va pas dans les zones qui lui sont interdites ou impossibles d'accès. Cette hypothèse est une représentation exacte de la physique du déplacement et du comportement du piéton dans l'espace en intérieur. Dans le cas du PF, il se trouve qu'elle est triviale à implanter.

Pendant l'étape de prédiction présentée en section 5.2.2, on ajoute une méthode qui, à l'itération t , à partir des N particules \mathcal{P}_t , et des N particules prédites $\tilde{\mathcal{P}}_t$, assure que pour toute particule E^i de position P^i et sa prédiction \tilde{E}^i de position \tilde{P}^i , le vecteur $\overrightarrow{P^i\tilde{P}^i}$ n'intersecte aucun mur et que \tilde{P}^i soit dans une position valide de la carte.

Dans ce cas, l'espace des hypothèses est rogné. Mathématiquement, on dit que le rognage de l'univers des variables aléatoires du processus stochastique par les segments de la carte forme un *support* à l'itération t . Topologiquement, ce support est défini comme un ouvert borné par une frontière polygonale.

Les particules invalidées par le filtrage par murs sont supprimées. Le nombre de particules dans le PF diminue. Il est nécessaire de remplacer les particules détruites. La régénération est déployée dans ce but.

5.3.3 Régénération

Elle consiste à régénérer les particules manquantes pour que l'intégralité des particules représente au mieux l'espace des hypothèses mis à jour. La zone d'optimalité du PF est sensible au nombre de particules jugé nécessaire pour que le PF puisse représenter correctement l'espace des hypothèses. Comme le filtrage par murs peut avoir supprimé certaines particules, il ne reste que \tilde{N} particules, tel que $\tilde{N} \leq N$. Le PF régénère les $N - \tilde{N}$ particules manquantes. La manière de régénérer ces particules affecte la façon dont le PF pave l'espace des hypothèses et ce qu'il « croit » être la position la plus probable. Si toutes les particules avaient le même poids, les particules se régénéreraient selon un vote majoritaire. Mais, nous proposons dans la section 5.4 une méthode pour pondérer les particules et introduire la vraisemblance issue des contraintes de la carte.

L'association du filtrage par murs et de la régénération émule un ré-échantillonnage où les particules atterrissant dans des zones inaccessibles (par exemple, derrière les murs) sont éliminées, et seules les particules valides auront une progéniture [Ghaoui et al., 2022].

Afin d'atteindre N particules, M nouvelles particules doivent être générées à partir de M sous-ensembles sélectionnés aléatoirement parmi \tilde{N} particules. \tilde{N} est défini comme $\tilde{N} = \lfloor ss \times \tilde{N} \rfloor$ où $ss \in]0, 1]$ est la taille proportionnelle du sous-ensemble.

L'algorithme 1 présente la fonction de régénération en détail. Cette fonction prend en entrée les `particules` valides résultants d'un filtrage par les murs de la carte et `ss` la taille relative du sous-ensemble. Elle renvoie en sortie l'ensemble de particules `régénérées`.

Pour chaque particule manquante, un sous-ensemble aléatoire parmi `particules` est sélectionné. Un exemple de tirage de ce sous ensemble peut être visualisé dans la figure 5.9. Les moyennes et variances de cet ensemble sont calculées. La particule moyenne de l'ensemble est enregistrée dans `moyennes` pour être testées plus tard. Une nouvelle particule est tirée en

supposant que les 4 composantes de celles-ci proviennent de distributions gaussiennes avec les moyennes et variances du sous-ensemble. La particule est enregistrée dans `candidates`.

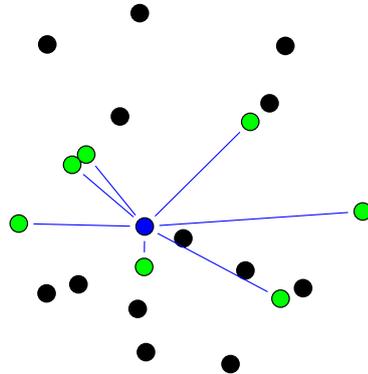


FIGURE 5.9 – En vert, l'ensemble de \widehat{N} particules sélectionnées pour la régénération.

Une fois que tous les tirages sont effectués, la fonction doit vérifier que les `candidates` ne traverse aucun mur par rapport à la particule moyenne du sous ensemble. Les `candidates` répondant à ce critère sont retournées pour être ajoutées aux particule du PF. Cette fonction est rappelée jusqu'à ce que $N = \widetilde{N}$.

Algorithme 1 : Fonction de régénération de particules.

```

Entrées : particules, ss
Sortie : régénérées
1 moyennes =  $\emptyset$ 
2 candidates =  $\emptyset$ 
3 pour  $i \leftarrow 1$  à  $N - \widetilde{N}$  par 1 faire
    // sélectionner sans répétition  $[ss \times N]$  particules
4     sous-ensemble = tirage aléatoire(particules, ss)
    //  $\theta$ , x , y and poids, moyens
5     moyennes.ajouter_fin(moyenne(sous-ensemble))
    // générer une nouvelle particule en calculant les moyennes et variances du
    sous-ensemble
6      $\mathcal{N}_\theta, \mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_w =$  propriétés_statistiques(sous ensemble)
7     candidates.ajouter_fin(particule( $\mathcal{N}_\theta, \mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_w$ ))
    // vérifier si les candidates intersectent un mur.
8 régénérées = filtrer_mur(candidates, moyennes)
9 retourner régénérées

```

Cette fonction de régénération par sous-ensemble est ré-entrante i.e. elle ne produit aucun changement en mémoire en dehors de son contexte local. Autrement dit, les particules régénérées n'affectent que l'appel suivant de la fonction et non pas l'appel actuel. Ceci implique aussi que la fonction est systématiquement parallélisable.

5.3.4 Sortie du filtre

Après chaque cycle d'estimation, le PF doit fournir une position moyenne 2D et un cap moyen. Ceci est fait car, un nuage de particules ou une distribution peuvent ne pas avoir un sens évident pour l'utilisateur, mais aussi, dans le cas où on veuille calculer une grandeur

telle que l'erreur de positionnement ou à des fins de visualisation. L'équation (5.25) permet de calculer la moyenne pondérée.

$$\bar{E}_t = \left[\sum_{i=1}^N w_t^i x_t^i \quad , \quad \sum_{i=1}^N w_t^i y_t^i \quad , \quad \sum_{i=1}^N w_t^i \theta_t^i \right]^T \quad | \quad \sum_{i=1}^N w_t^i = 1 \quad (5.25)$$

Dans le cas où le filtre est en mode multi-hypothèses, l'équation (5.25) devient :

$$\bar{E}_t = \frac{1}{N} \left[\sum_{i=1}^N x_t^i \quad , \quad \sum_{i=1}^N y_t^i \quad , \quad \sum_{i=1}^N \theta_t^i \right]^T \quad (5.26)$$

5.3.5 Sortie du filtre contrainte

Dans la section 5.3.4, le filtre a pour sortie la moyenne pondérée de ses particules. Sauf que ce processus trivial présente un cas limite. On suppose que les particules sont distribuées sur un virage comme dans la figure 5.10. Dans ce scénario, il est possible que la moyenne pondérée (particule verte) résulte derrière le mur. Ce qui est physiquement impossible.

Nous proposons une solution qui vise à trouver la particule valide la plus proche au sens de la distance euclidienne à la moyenne pondérée. Cette particule est désignée comme la position estimée par le filtre après ce pas. Cette solution ajoute un temps de calcul de complexité $\mathcal{O}\{N\}$ mais est nécessaire au fonctionnement correct du PF et évite d'effectuer une recherche de domaine de validité.

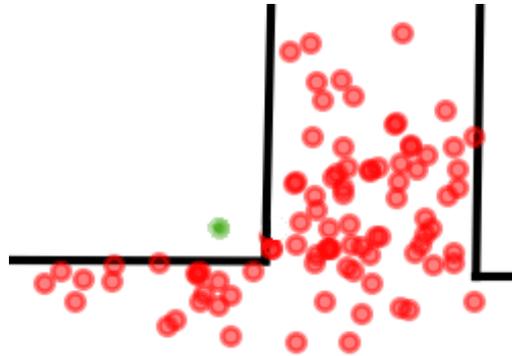


FIGURE 5.10 – Particules distribuées autour d'un virage.

5.4 Filtrage par modèle de mouvement humain

Pour représenter le mouvement humain à l'intérieur d'un bâtiment sur un plan d'étage, nous proposons d'introduire un *a priori* sur la vraisemblance lors du mouvement associée aux différentes dispositions et configurations intérieures. Nous supposons qu'un humain est plus susceptible de se déplacer au centre d'un couloir plutôt que de longer le mur, de traverser le centre d'une porte plutôt que de coller aux côtés. Cet *a priori* est représenté sous la forme d'une grille.

5.4.1 Grille de vraisemblance

Une grille g est une structure de données à 2 dimensions minimum définie dans $\mathbb{N}^{r \times c}$ avec r et c le nombre de ligne et de colonne de cette grille. Les grilles sont une représentation discrète d'un espace continu où une case (élément) est identifiée par ses indices. Chaque case détient l'information associée à sa position dans la grille et donc dans l'espace de localisation. Une case peut évidemment contenir plusieurs informations de natures différentes. Cette grille à pour but de répondre à l'objectif [C].

Le choix de la grille est lié au fait que le processus de localisation vise à estimer où le pied d'un piéton atterrit. Une grille est un compromis satisfaisant entre : finesse de représentation si la taille des cases est comparable à la dimension minimale du pied ; vitesse de calcul ; et impact mémoire. Ce compromis doit satisfaire l'objectif [F] de cette thèse qui est lié au téléphone. Tout comme le graphe, la grille exprime les relations de voisinage. Mais, ces relations sont systématiques i.e. les voisins en 4 ou 8 adjacence sont connus via l'addition ou la soustraction des indices. Comparée au graphe, la grille a donc une empreinte mémoire fixe après sa construction et peut être pré-calculée. Enfin, vu que chaque élément est accédé par des indices, le temps d'accès à cet élément est de complexité $\mathcal{O}\{1\}$.

Nous convertissons la carte d'étage de longueur L et de largeur W en une grille : $G \in \mathbb{R}^{l \times w}$ de longueur $l = L/scale$ et de largeur $w = W/scale$, avec chaque case $g \in G \mid g \in [0, 1]$. Dans notre cas, $scale = 5 \text{ cm}$. La grille est donc composée de cases de $5 \times 5 \text{ cm}^2$. Après l'étape de discrétisation, la carte des distances de chaque case au mur le plus proche est construite.

5.4.2 Fonctions d'appartenance

Il est nécessaire de transformer la carte des distances obtenue en grille de vraisemblance dont les valeurs sont comprises entre 0 et 1. Pour cela, une fonction d'appartenance f est définie comme :

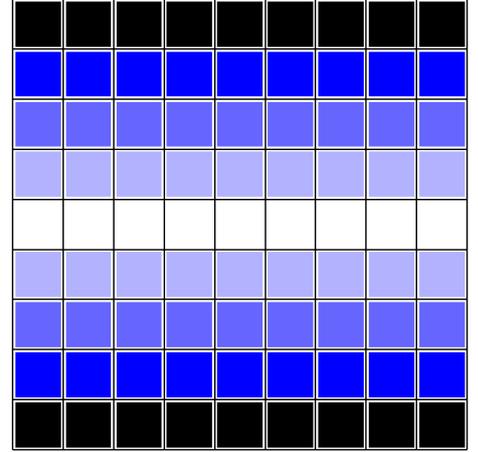
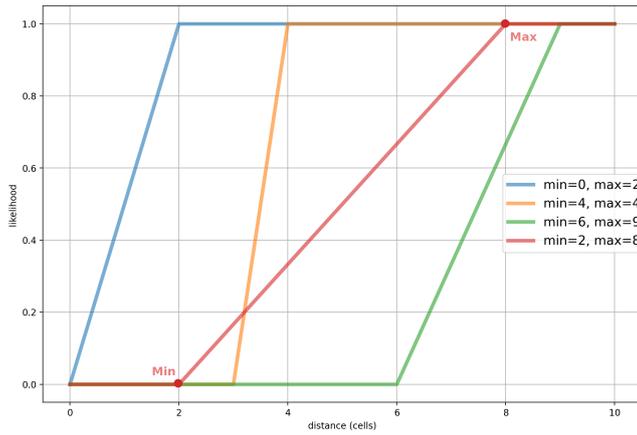
$$f : \mathbb{N}^{l \times w} \rightarrow [0, 1]^{l \times w} \quad (5.27)$$

Le choix de la fonction d'appartenance, illustrée sur la figure 5.11a à titre d'exemple, affecte la manière dont la grille contraint les particules lors l'étape de mise à jour. Dans tous les cas, la fonction f crée une contrainte de centrage car elle pousse les particules au centre du couloir.

Ces fonctions d'appartenance ont 3 zones et 2 paramètres :

- $d \in [0, \mathbf{min_range}]$, zone interdite.
- $d \in [\mathbf{min_range}, \mathbf{max_range}]$, zone de transition.
- $d \in [\mathbf{max_range}, \infty]$, zone autorisée.

La grille résultante est illustrée à la figure 5.12 et représente notre a priori sur la vraisemblance du mouvement humain. Un exemple zoomé d'une grille de corridors est donné à la figure 5.11b. Les cases noires sont des murs avec une probabilité associée de 0. Les cases plus claires ont une vraisemblance plus élevée qu'un pas se produise sur elles. L'équation d'état 5.17 montre que le poids des particules est multiplié par cette vraisemblance. Le choix des paramètres **min_range** et **max_range** est expliqué dans l'annexe B.



(a) Exemple de fonctions distance - vraisemblance $f(d) = \mathcal{L}$. (b) Exemple de grille de couloir.

FIGURE 5.11 – Concepts de grille.

5.5 Expériences

Les expériences suivantes ont pour but de produire des tests unitaires afin de valider le fonctionnement du filtrage via les murs de la carte et la régénération en section B.2.1. Ensuite, les expériences concernant l'étude de l'effet du nombre de particules en section 5.5.1, l'effet du gain de *Weinberg* en section 5.5.2, la réalisation d'une trajectoire par le PF en section 5.5.3 et l'introduction de la grille de vraisemblance sont présentées en section 5.5.4. Une comparaison est proposée en section 5.5.4.1.

Avant propos : le PF exploite les variables aléatoires δ et α de l'équation (5.17) lors de l'estimation. Ces valeurs étant tirées aléatoirement, les expériences présentées ci-dessous doivent tenir compte de ce caractère aléatoire en procédant à un tirage par la méthode de Monte Carlo.

Une expérience est exécutée de manière identique dix fois. Donc, les données, paramètres, algorithmes et mesures sont figés. Seuls les tirages des nombres aléatoires changent. Ces tirages dépendent du générateur de nombre pseudo-aléatoire (RNG). Cet RNG possède une "graine" qui est modifiée pour chaque tirage. Cette méthode est déployée dans le but d'éviter de biaiser les résultats des expériences et d'avoir une idée plus précise de l'impact des variations d'un paramètre. Car, il est effectivement possible qu'une séquence de tirage soit "favorable" ou "défavorable" faisant paraître des erreurs mesurées plus faibles ou plus importantes, respectivement. Ce sont donc les moyennes et écart-types des mesures sur les dix simulations qui sont présentés à chaque fois que PF est concerné par une expérience.

Les tests de validations des différentes étapes du PF sont présentés en annexe B. Ils démontrent le fonctionnement d'un mécanisme que nous avons implanté et permettent la visualisation de celui-ci dans un scénario de localisation réduit ou simplifié. Ils sont une nécessité afin de valider les différentes étapes du filtre pour pouvoir mener les études présentées ci-dessous.

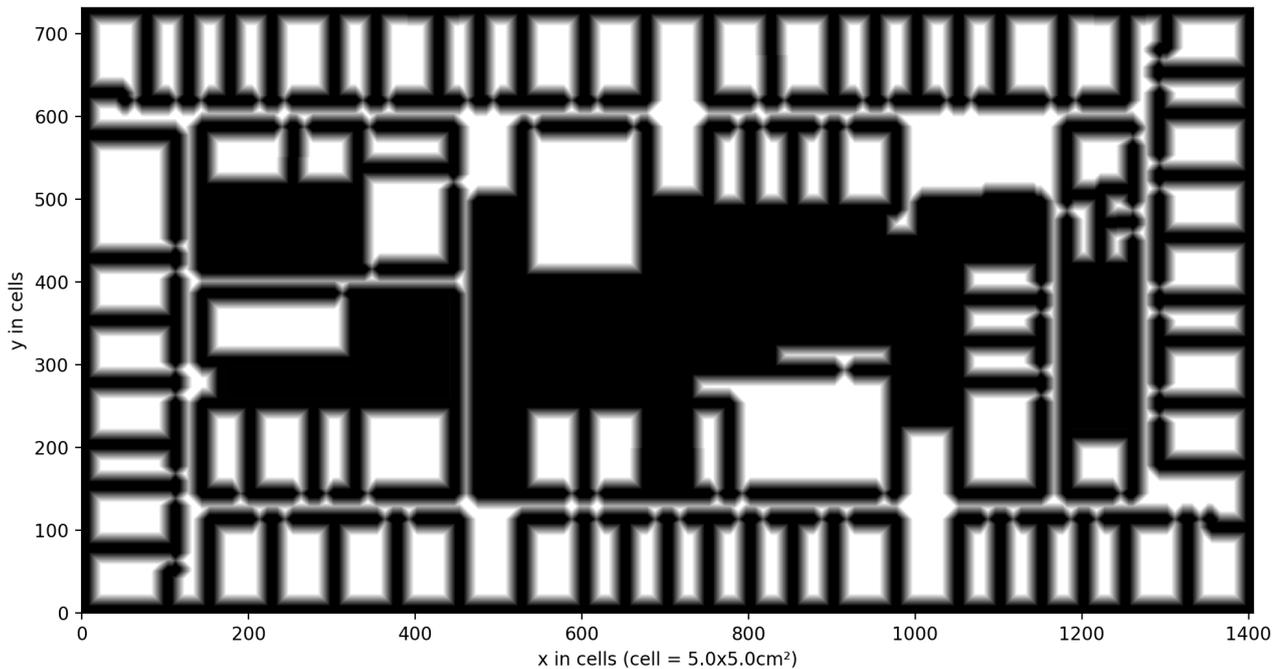


FIGURE 5.12 – Grille de vraisemblance construite à partir du plan de la carte avec **min_range**=8 cell et **max_range**=12 cell.

5.5.1 Étude de l'effet du nombre de particules

Dans la section 2.5.1.4, il est mentionné que le choix du nombre de particules nécessaires pour le fonctionnement du PF n'est pas explicité dans l'état de l'art. Il faut une méthode pour déterminer ce nombre. L'expérience suivante est proposée :

1. Établir la vérité terrain par la procédure proposée dans la section section 4.3.
2. Parcourir la trajectoire planifiée en enregistrant les instants de passage sur les points repères.
3. Estimer la trajectoire à partir des données collectées en utilisant un nombre de particules N .
4. Calculer les distances euclidiennes et Mahalanobis aux points repères par rapport à la trajectoire estimée.
5. Répéter depuis 3 pour une valeur différente de N .

Le but est d'observer l'impact du nombre de particules N sur les moyennes et écart-types des distances euclidiennes et Mahalanobis pour les mêmes données, paramètres, trajectoires et vérité terrain.

Dans la figure 5.13, les valeurs moyennes des distances euclidiennes et Mahalanobis diminuent en fonction du nombre de particules. Le système converge pour au moins 200 particules où les écart-types des moyennes des deux distances englobent toutes les valeurs pour $N > 200$ particules. Au-delà de ce nombre, il n'y a pas de changement significatif sur une des mesures ni sur l'autre. Mais, le coût de calcul et les ressources informatiques nécessaires augmentent. Nous utilisons 200 particules pour les expériences restantes.

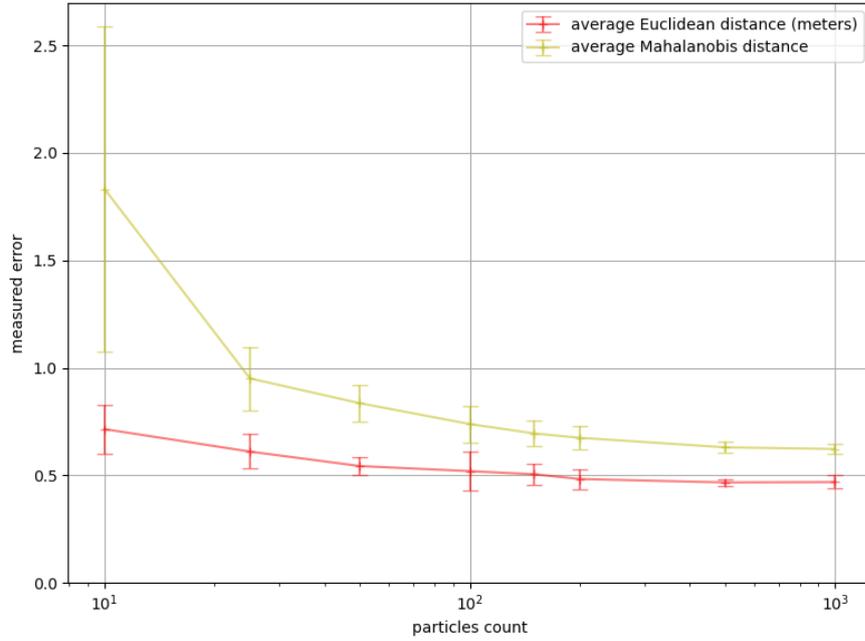


FIGURE 5.13 – Évolution des erreurs mesurées en fonction du nombre de particules [Ghaoui et al., 2022].

5.5.2 Étude de l’effet du gain de Weinberg

Dans la section 4.1.3.1 le gain de *Weinberg* est évalué par pré-calibration pour un seul utilisateur. Le test mené a présenté une mésestimation de ce gain. Dans l’expérience suivante, le même piéton parcourt une trajectoire dans un sens puis dans l’autre. Les configurations et les paramètres sont identiques dans les deux cas. Seul l’ordre de passage sur les points repères change. Les distances euclidiennes et Mahalanobis sont comparées pour les trajectoires pour différents gains de *Weinberg*. Les taux de succès de retour à la position de départ sont aussi comparés sur la figure 5.14.

La figure 5.14 montre que E et M (courbes rouges et jaunes) atteignent une valeur minimale proche de la valeur empirique du gain de Weinberg (1.07). Les courbes bleue et cyan pour la trajectoire en sens inverse montrent que E et M atteignent une valeur minimale différente.

Les trajectoires n’ont pas de valeur optimale de gain de *Weinberg*. Néanmoins, il existe un intervalle où les taux de réussite se chevauchent et maintiennent les erreurs à une valeur minimale. Le tableau 5.2 montre la comparaison entre la valeur du gain de *Weinberg* trouvée empiriquement et calculée statistiquement. Cela implique que le bruit de pas δ corrige les erreurs dans les estimations de longueur de pas.

Gain de <i>Weinberg</i> de l’utilisateur 1	empirique	statistique
forward	1.07	[1.05, 1.09]
reverse	1.07	[1.05, 1.110]

TABLE 5.2 – Gains de *Weinberg* empirique et statistique.

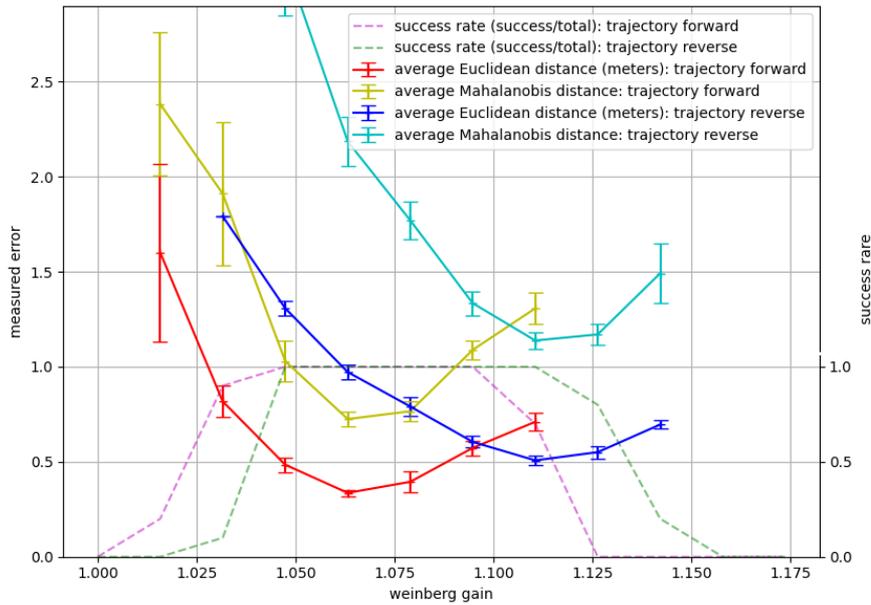


FIGURE 5.14 – Évolution des distances mesurées en fonction du gain de Weinberg.

5.5.3 Trajectoire réalisée

Ce test consiste à estimer la trajectoire qu'un piéton effectue dans le bâtiment via le PF. Le point de départ et d'arrivée sont les mêmes. La trajectoire fait environ 200 m de long. Elle contient différents passages à travers des couloirs, passages semi-ouverts et aucune boucle.

Dans la figure 5.15, la trajectoire en bleu est estimée par PF. Chaque pas détecté peut être distingué. Chaque point bleu est considéré comme la moyenne pondérée du PF contraint comme expliqué dans la section 5.3.5. Cette trajectoire présente une erreur euclidienne moyenne 0,55 m et une erreur Mahalanobis moyenne de 1,1. Les deux corrections proposées par le PF sont entrées en action : la distance estimée pour un pas est corrigée par le tirage des valeurs du bruit de propagation sur la longueur de pas δ ; le cap est corrigé par le bruit de propagation en angle α , qui élargit le cône servant de support pour les particules, et le filtrage par les murs de la carte qui élimine les particules ayant enfreint une contrainte de déplacement. Cette trajectoire est une réalisation de la succession des tirages des variables aléatoires qui ont servi à estimer la position et le cap du piéton.

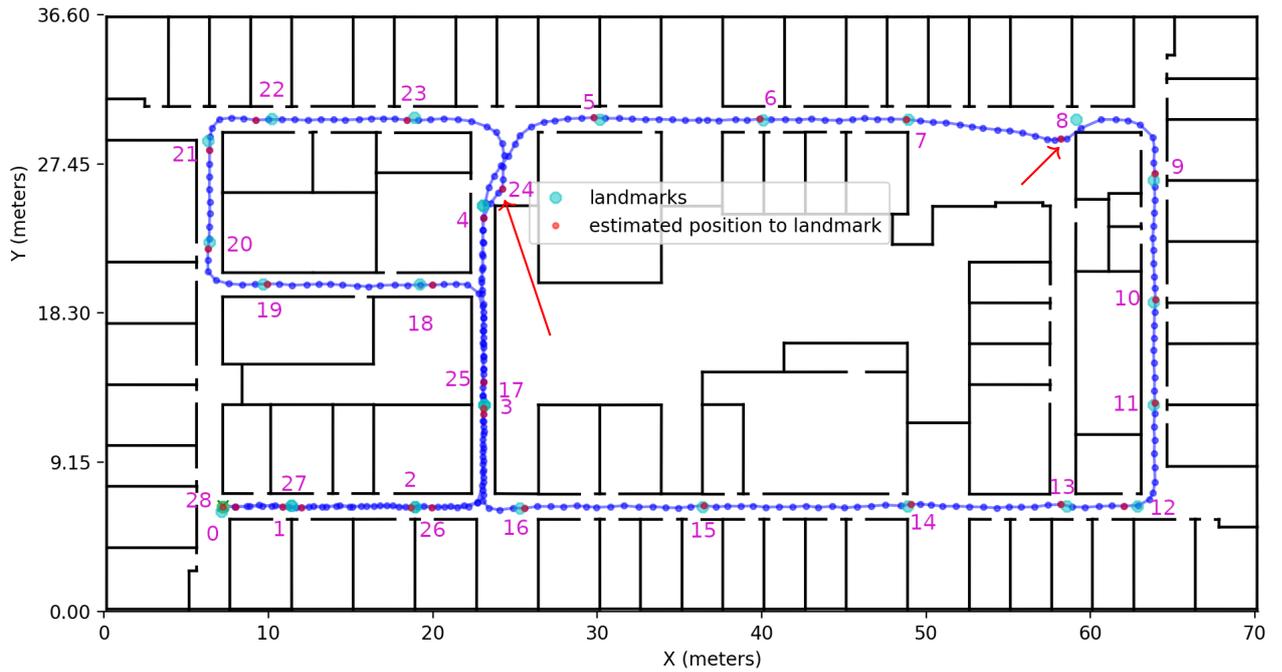


FIGURE 5.15 – Trajectoire d'un piéton estimée par le PF. Chaque point bleu représente un pas. Les points cyans sont les points repères. les points rouges sont les positions estimées à l'instant de la mesure [Ghaoui et al., 2022].

En observant la trajectoire au point 8 et 24, elle semble être biaisée. Afin de comprendre ce résultat, on visualise toutes les particules qui ont servies à produire cette réalisation dans la figure 5.16. Ceci permet de comprendre comment le filtre a représenté l'espace des hypothèses tout au long de la trajectoire et des différentes configurations de l'espace en intérieur.

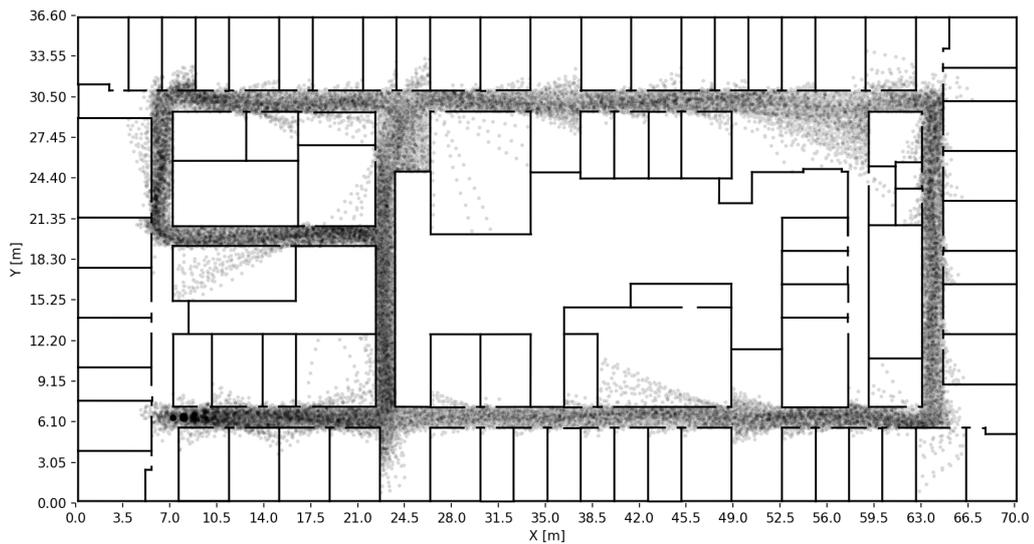


FIGURE 5.16 – Représentation de toutes les particules de la trajectoire de la figure 5.15.

On remarque que toutes les particules ont le même poids car elles ont le même niveau de gris

sur la figure. L'intensité du niveau de gris dépend de la superposition de plusieurs particules. Ceci est cohérent avec le formalisme adopté en section 5.2.3 et donc le filtre fonctionne en mode multi-hypothèses dans ce cas. On note aussi que le point de départ $\approx (7\text{ m}; 6, 10\text{ m})$ contient un regroupement noir dense. Celui-ci est lié à l'initialisation localisée présentée en section 5.2.1. Il est important de noter que des particules passent par des endroits faux e.g. à l'intérieur des salles. Les particules se propagent uniformément dans les espaces ouverts. Mais ce n'est pas le cas dans les espaces semi-ouverts. Les particules d'un côté du plan sagittal sont supprimées car elles entrent en collision avec le mur alors que les autres survivent. Le nuage de particules dérive donc loin du mur jusqu'à rentrer dans un couloir qui rééquilibre la distribution des particules.

5.5.4 Effet de la grille de vraisemblance

Dans cette expérience, la même configuration que celle de l'expérience précédente est utilisée à la différence que le grille de vraisemblance présentée en section 5.4 est utilisée. Le PF modifie le poids des particules en fonction de leur position dans cette grille.

Dans un premier temps, la trajectoire obtenue est la même que celle présentée en figure 5.15 avec une différence sur l'erreur moyenne non-significative. Donc à première vue, la grille n'a pas d'impact sur la précision de localisation. Cependant, en observant l'évolution du nuage de particules dans la figure 5.17 comparée à figure 5.16, les particules n'occupent que très rarement une zone de la carte que le piéton n'a pas visitée i.e. aucune salle.

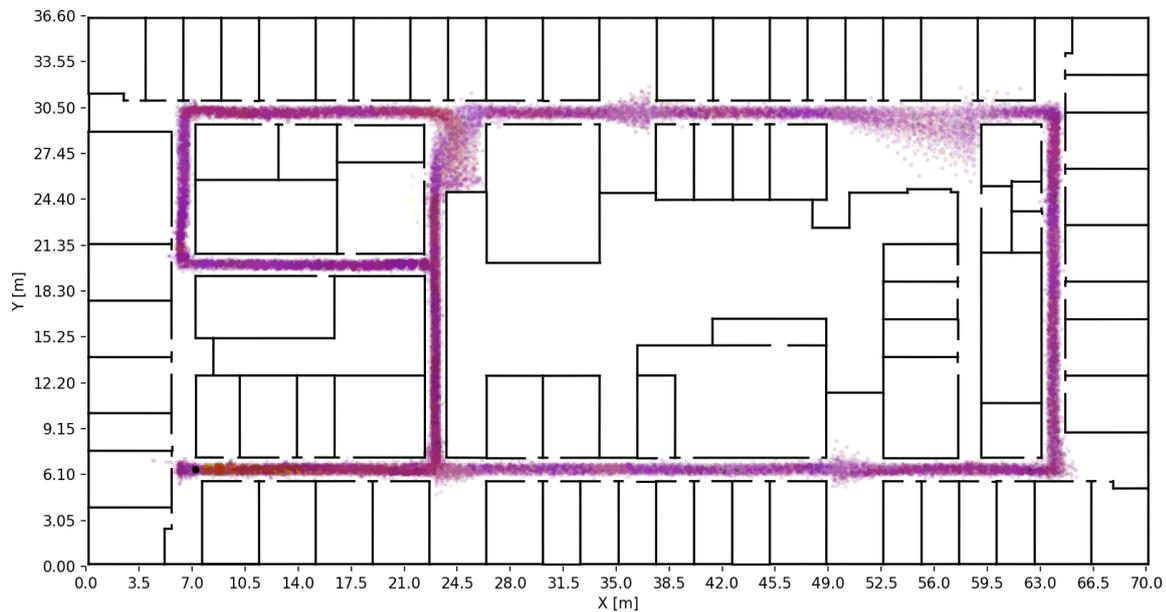


FIGURE 5.17 – Représentation de toutes les particules de la trajectoire.

Cela implique que la grille possède un impact sur la distribution des particules. Afin d'examiner cet impact, nous proposons de visualiser les ellipsoïdes d'incertitude à 3σ autour de chaque positions moyennes pour les 29 points de mesure de la figure 5.15. Les figures 5.18 et 5.19 sont proposées pour une comparaison visuelle. Nous remarquons que les ellipsoïdes sont contraints par les murs et les traversent rarement lorsque la grille est activée. Cela montre que la grille a un effet amélioratif sur la localisation pour ces configurations intérieures. Les

couleurs des particules indiquent leur poids. Dans le cas de la grille désactivée, toutes les particules ont le même poids (couleur) $1 / N$.

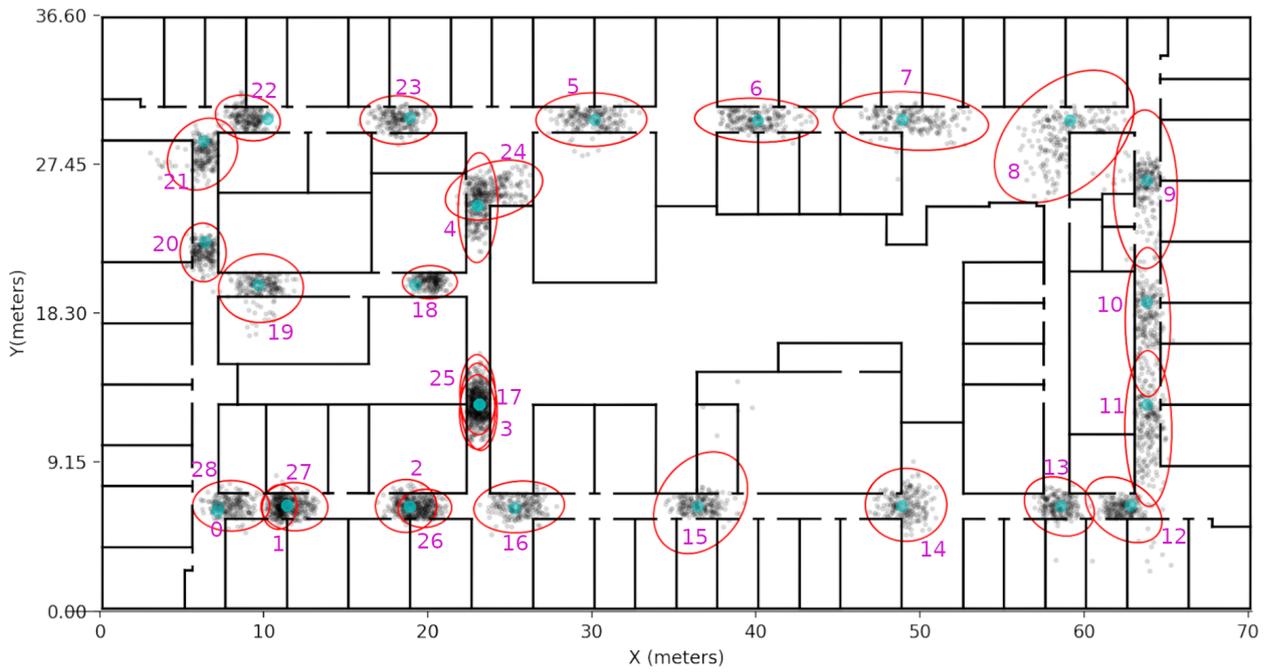


FIGURE 5.18 – Ellipsoïdes d’incertitude visualisés sans la grille de vraisemblance.

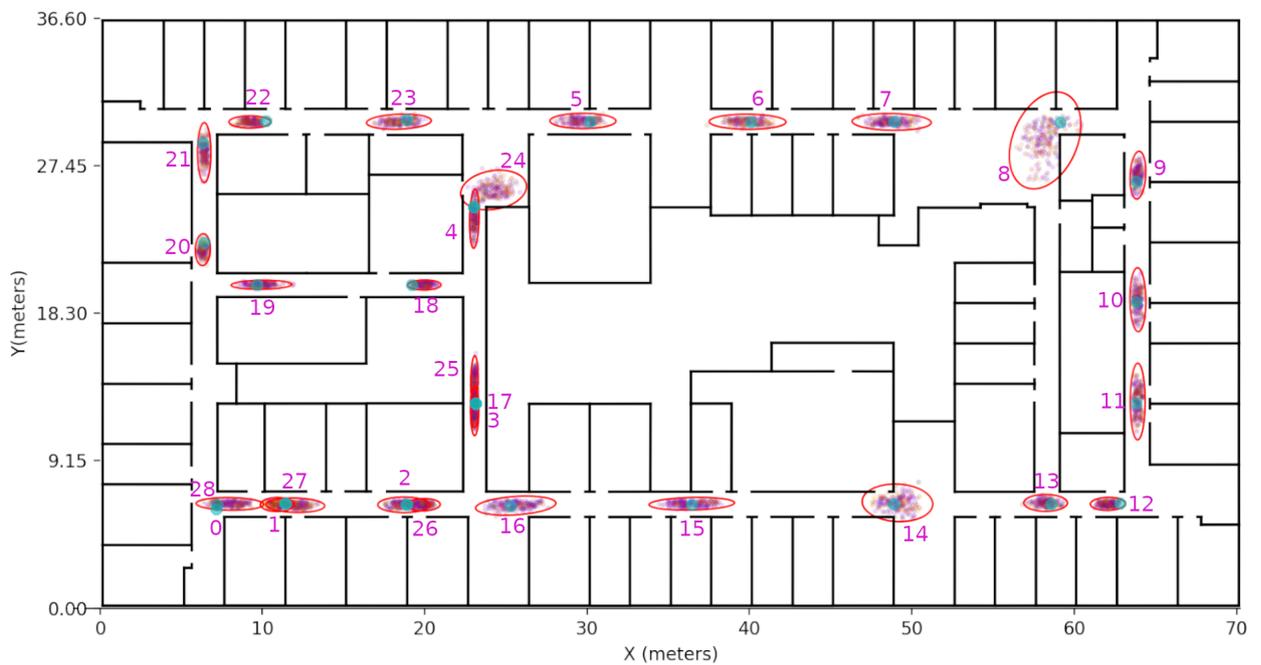


FIGURE 5.19 – Ellipsoïdes d’incertitude visualisés avec la grille de vraisemblance.

Le comportement du PF est intéressant aux points 8, 12, et 24 de la figure 5.15. En observant la figure 5.19, on remarque que le repère 8 vient après une marche de ≈ 10 m

dans un espace semi-ouvert. Ainsi, les particules non contraintes ont tendance à se propager et à paver plus largement les espaces d'hypothèses. Le repère 12 montre une compression intéressante de l'ellipsoïde des particules. Il vient après un virage à 90° avec un espace semi-ouvert, mais la version avec la grille restreint l'ellipsoïde et n'a pas de particules dans les pièces adjacentes comme le repère 9 sur figure 5.18. Le repère 24 est un cas limite car il combine les deux difficultés de 8 et 12. Il s'agit d'un espace semi-ouverte après un virage à 90° . Les particules se propagent facilement pour remplir l'espace. Ensuite, elles sont canalisées dans le couloir étroit. Cela fait que l'ellipsoïde s'étend à son maximum puis se replie dans le couloir. Enfin, notons aussi que pour les points 8 et 12, il existe une dérive liée à un biais dû à la présence du mur car les particules ayant touché le mur ont été supprimées et celles qui lui sont proche, reçoivent un poids faible voire nul. Donc, l'espace des possibles dévie en s'éloignant du mur. Ce phénomène sera corrigé par l'introduction d'un signal extéroceptif et sera présenté dans la section 8.1. Tous ces phénomènes physiques sont correctement représentés par des pics d'erreur E_t et d'écart-type σ_t sur les deux axes X et Y sur la figure 5.20.

Dans l'optique d'examiner l'intégrité de la localisation et de montrer l'impact de la grille de vraisemblance sur celle-ci, nous proposons la figure 5.20. Les courbes présentées montrent les erreurs moyennes signées en position sur l'axe X et Y en rouge pour la version avec grille et en bleu pour la version sans grille. Ces erreurs sont entourées par les courbes de leur écart-type ($\pm 3\sigma$). Il est dit que le système est intègre si l'asymptote $y=0$ est contenu dans le couloir emprisonnée entre -3σ et $+3\sigma$. Cela se traduit par le fait qu'à chaque instant de mesure, dans 99.7% des cas, la véritable position du piéton est incluse dans l'ellipsoïde d'incertitude, mais aussi, le fait que le couloir $\pm 3\sigma$ avec grille (magenta) soit plus étroit que celui sans grille (cyan). La grille de vraisemblance permet de renforcer l'intégrité du système et de réduire l'incertitude de positionnement répondant à l'objectif [C].

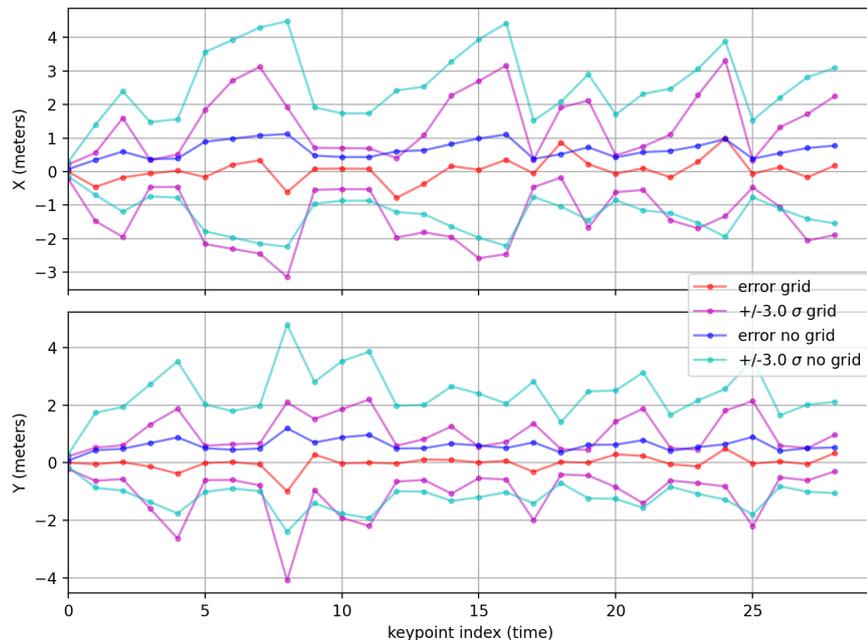


FIGURE 5.20 – Erreurs signées mesurées $\pm 3\sigma$ à chaque point repère.

5.5.4.1 Comparaison avec PDR et RoNIN

Le système présenté utilisant un filtre particulaire est comparé avec une réalisation du PDR présentée en chapitre 4 section 4.1 et au modèle d'intégration robuste basé sur des réseaux de neurones RoNIN [Yan et al., 2019]. Dans la figure 5.21, la comparaison de 3 méthodes : notre proposition en bleu, une navigation piétonne à l'estime (PDR) en orange, l'estimation par RoNIN en magenta.

La trajectoire résultante montre que le filtrage par les murs de la carte est obligatoire pour les environnements intérieurs car il représente le déplacement physique plus fidèlement que les méthodes statistiques pures. Il est également important de noter que RoNIN s'appuie sur l'orientation d'Android (*Game Rotation Vector*) qui elle-même exploite les données défectueuses du magnétomètre en espace intérieur. Pour montrer plus en détail la différence entre les 3 approches, le tableau 5.3 donne l'erreur euclidienne calculée pour chaque système.

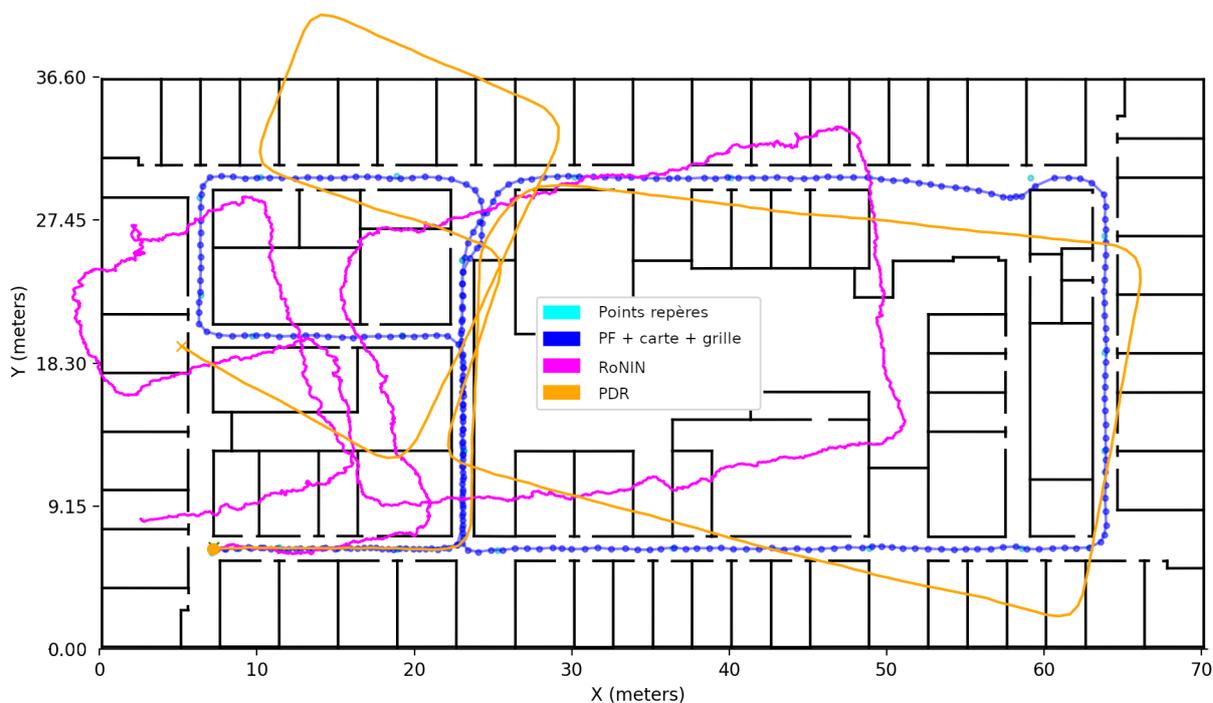


FIGURE 5.21 – Comparaison entre les estimations du PF, RoNIN et PDR.

Système	erreur [m]
PF + carte+ grille ■	0.811
RoNIN ■	7.534
PDR ■	4.671

TABLE 5.3 – Résultats numériques de la comparaison de figure 5.21

5.6 Récapitulatif

La correction de la trajectoire d'un piéton est un problème d'estimation bayésienne récursive. Le piéton est représenté par un vecteur d'état à trois composantes. Le but est d'estimer l'état suivant en utilisant l'état actuel et les informations provenant des méthodes présentées en section 4.1. Il est admis que dans la plupart des cas les équations de récurrence de ce problème n'ont pas de solutions sans introduire plus d'hypothèses. Le filtre particulaire (PF) est un formalisme qui permet de résoudre ce problème d'estimation en approximant la distribution *a posteriori* par un ensemble fini de particules. Le PF est adapté aux problèmes d'estimation d'état *a posteriori* non linéaire non gaussien.

Le filtre à particules modélise l'espace des possibles et suit son évolution au cours du temps/événements avec les N particules. Il considère que la physique d'un problème peut être modélisée par un processus stochastique. Ses particules représentent des hypothèses sur les trajectoires possibles que le piéton a pu prendre pendant un pas. La même information de longueur et cap de pas est transmise aux particules. La différence est apportée par chaque particule qui réalise un tirage de variables aléatoires sur : la longueur du pas δ et le cap du pas α . L'équation régissant le PF est l'équation (5.17).

Le PF est découpé en quatre étapes essentielles : initialisation, prédiction, mise à jour, ré-échantillonnage. Nous avons présenté chacune d'entre elles ainsi que les choix d'instanciation pris pour cette thèse. Pour l'initialisation, nous avons opté pour une initialisation localisée où la position et le cap de départ sont représentés par des distributions gaussiennes autour des véritables position et cap de départ dans le but de renforcer la robustesse du système. La prédiction s'effectue de manière standard où toutes les particules reçoivent une longueur de pas SL_t et une différence de cap $d\theta_t$. Chaque particule ajoute alors son tirage des variables aléatoires δ et α qui sont tirées de distributions gaussiennes présentées dans le tableau 5.1. L'étape de mise à jour est celle qui interfère avec le poids des particules afin d'incorporer une information observée ou un *a priori*. Cette étape peut mener au phénomène de dégénérescence du filtre où une majorité de particules a un poids négligeable et en même temps quelques particules s'accapare le poids. Le filtre échoue alors à représenter l'espace des possibles correctement car le nombre de particules significatives est réduit. Il faut donc procéder à la quatrième étape : le ré-échantillonnage où les particules les plus faibles sont supprimées, et les particules les plus importantes sont dupliquées. Notons que ce processus peut lui même conduire à un phénomène nommé "appauvrissement de échantillons" où un amas de particules aux poids forts se regroupent et éclipsent toutes les autres. Il faut donc établir un critère de déclenchement du ré-échantillonnage. Nous avons en avons présenté deux : [Kong et al., 1994] et [Pham et al., 2003].

Nous avons proposé le filtrage par carte et la régénération pour remplacer le ré-échantillonnage. La carte de l'étage du bâtiment sert à appliquer les contraintes spatiales sur le mouvement humain : l'impossibilité d'accéder à des parties restreintes du bâtiment ou de franchir un mur pour une particule. Cela invalide la particule et détruit donc cette hypothèse car cette trajectoire est impossible. Après ce filtrage, le nombre de particules est inférieur à celui nécessaire au fonctionnement correct du PF. Il est nécessaire de régénérer les particules manquantes. Pour cela, nous avons proposé un processus de régénération par échantillonnage aléatoire des sous ensembles de particules survivantes. Notre régénération tient toujours compte des contraintes spatiales où une particule ne peut pas être régénérée dans une position qui n'est pas en accord avec le sous ensemble aléatoire sélectionné. Cette combinaison de filtrage et régénération est

présente dans notre publication [Ghaoui et al., 2022].

Par la suite, nous avons présenté le calcul de la moyenne pondérée des positions et caps des particules. Initialement, ce calcul semble trivial mais il présente un cas limite où la moyenne des particules peut résulter dans une position qui enfreint les contraintes spatiales. Le cas type est présenté par la figure 5.10.

Nous avons aussi proposé la grille de vraisemblance pour pondérer les particules lors de la mise à jour. Cette grille est présentée en section 5.4 et permet la représentation de la vraisemblance du mouvement humain dans les espaces intérieurs ainsi que de décrire des comportements de déplacements du piéton en accord avec l’objectif [H]. Cette grille fait partie de l’une de nos contributions publiées dans [Ghaoui et al., 2022]. Nous avons choisi la représentation de cette vraisemblance sous la forme d’une grille car une grille est un compromis satisfaisant entre finesse de représentation si la taille des cases est comparable à la dimension minimale du pied ; temps d’accès constant ($\mathcal{O}\{1\}$) et impact mémoire calculable, dans notre cas 1 Mo pour un étage faisant 70 m \times 36 m de surface. Ces temps de calcul et impact mémoire sont compatibles avec l’objectif [F] pour le téléphone.

Nous avons aussi présenté via nos expériences la validation du processus de filtrage (annexe B) par les murs de la carte et l’ajout de la régénération afin d’éviter que la localisation par le filtre échoue dans le cas où toutes les particules sont supprimées. Ce test a conclu que les deux processus incarnent une représentations fidèle de la physique du mouvement et qu’ils ont donc un impact amélioratif très conséquent sur la trajectoire estimée.

Le nombre de particules nécessaires au fonctionnement correct du filtre a été étudié afin de répondre à la question que nous posons dans la section 2.5.1.4 sur comment déterminer ce nombre. L’évolution des métriques mesurées en fonction de ce nombre indique qu’à partir de 200 particules, le système n’a pas de gain de performances significatif mais uniquement une augmentation du coût de calcul. Nous avons donc sélectionné $N = 200$ particules.

Une comparaison sur l’effet du gain de *Weinberg* sur une trajectoire dans un sens et dans l’autre a été présentée. Il s’avère que pour une trajectoire similaire, le gain de *Weinberg* n’a pas une valeur fixe par piéton. D’où la nécessité d’introduire le bruit de propagation sur la longueur de pas δ . Ce bruit sert à corriger une mésestimation du gain de *Weinberg* qui est en accord avec l’objectif [B].

La trajectoire obtenue par l’estimation du filtre particulaire sans l’usage de la grille de vraisemblance indique une erreur de localisation moyenne de 0,55 m. Quand la totalité des particules est observée, certaines d’entre elles se propagent dans des salles que le piéton n’a pas visitées. Nous avons ensuite introduit pour les mêmes données un traitement où la grille de vraisemblance est active. La grille a pour effet d’appliquer une contrainte aux particules de centrage dans un couloir. Cette contrainte fait que les ellipsoïdes sont resserrés dans les couloirs et que les particules couvrent une moins grande surface après être sorties des couloirs. En comparant l’incertitude autour de la position moyenne, nous notons que la version où la grille est active a une incertitude inférieure à celle où elle est inactive. Notre grille de vraisemblance permet donc de renforcer l’intégrité du système de localisation en assurant que dans 99,7% des cas la véritable position soit incluse dans l’ellipsoïde d’incertitude en accord avec l’objectif [C]. La pondération des particules est possible grâce à la vraisemblance tirée de cette grille.

Une dernière comparaison est proposée entre le système de localisation schématisé par la figure 5.21 utilisant un PF, les murs de la carte et la grille de vraisemblance ; un PDR du chapitre 4 section 4.1 ; et le modèle neuronal d’intégration robuste RoNIN [Yan et al.,

2019]. Les erreurs euclidiennes suivantes sont reportées : PF + carte+ grille : 0,811 m ; PDR : 4,671 m ; RoNIN : 7,534 m. Cela montre que l'usage des cartes a un impact très important pour un coût d'implantation faible à cause de leur disponibilité dans les espaces intérieurs.

Toutefois, l'introduction du filtrage par mur et de la grille de vraisemblance provoque l'apparition d'un biais sur les zones semi-ouvertes où les murs détruisent les particules invalides d'un côté de l'espace des possibles ; et la grille attribut des poids important aux particules de l'autre côté de cet espace. Ce phénomène est corrigé par l'introduction d'un signal extéroceptif et est présenté dans la section section 8.1.

5.7 Conclusion

Dans ce chapitre, nous avons présenté le filtre particulaire ainsi que les détails autour de ces étapes. Les modifications que nous avons apportées à ces étapes ont été exposées. Nous y introduisons aussi l'usage de la carte dans le PF. Cette carte permet de filtrer les particules invalides. Un processus de régénération accompagne ce filtrage pour compenser les particules invalides effacées. Notre contribution concernant la représentation de la vraisemblance du mouvement humain sous la forme d'une grille y est présentée. Nous expliquons comment cette grille est construite et montrons son impact sur le processus de localisation via une série d'expériences. Ces expériences ont permis de valider le fonctionnement des parties du filtre ainsi que d'analyser cet impact. Le récapitulatif de la partie I de ce manuscrit est présenté dans le chapitre 9.

Deuxième partie

Localisation assistée par vision

Chapitre 6

État de l’art de la vision par caméra en localisation

Sommaire

6.1	Caméra fixe	114
6.2	Caméra mobile	115
6.2.1	Caméras 2D vs 3D	115
6.2.2	Amers artificiels vs organiques	116
6.2.3	Localisation et cartographie simultanées (SLAM)	117
6.2.4	Réalités augmentée (AR), virtuelle (VR) et mixte (MR)	120
6.2.5	Traitement traditionnel vs intelligence artificielle	121
6.3	Détecteurs d’objets dans une image	122
6.3.1	YOLO	122
6.3.2	Concurrents de YOLO	125
6.4	Synthèse	126
6.5	Structure de la partie II	129
6.5.1	Contributions	129

Dans la première partie de ce manuscrit, la localisation proprioceptive est étudiée. Cette localisation présente déjà des résultats exploitables et permet de répondre à une parties des objectifs de ce manuscrit. Cependant, Elle connaît ses limites sur les espaces intérieurs dont la configuration est plus complexes et moins contrainte par la carte du bâtiment. Ceci appelle à l'usage d'une information extéroceptive pour compléter la localisation proprioceptive.

Dans le chapitre 2, il est mentionné que la localisation peut être effectuée en exploitant une ou plusieurs caméras. Il existe plusieurs approches à cette problématique et elles répondent différemment à la tâche demandée en fonction du nombre de capteurs utilisés, de leur homogénéité ou hétérogénéité et de leur positionnement relatif. Ce chapitre aborde les techniques connues pour déduire la position du piéton en utilisant ces caméras. Le but est d'intégrer une source de signal extéroceptif pour notre système résultant du chapitre 5.

6.1 Caméra fixe

Une première approche consiste à utiliser les cameras déjà en place i.e les caméras de surveillances. Une caméra voyant un piéton dans son champ de vision peut localiser celui-ci par une projection au sol. Il est nécessaire de calibrer ces caméras ou d'avoir une information supplémentaire via des capteurs portées sur le piéton. [Sun et al., 2016] ont proposé une solution de localisation utilisant des caméras panoramiques fixes et une carte de l'environnement intérieur. Ils ont appliqué une méthode de soustraction de fond pour détecter les êtres humains et faire correspondre leur emplacement à une position correspondante sur la carte intérieure.

[Desai and Rattan, 2009] ont utilisé une caméra fixe "*pan/tilt*" et des réseaux de capteurs sans fil pour suivre des objets dans un espace intérieur. L'estimation de la position d'un objet a été effectuée avec la méthode de la différence de temps d'arrivée. La caméra, équipée d'un pointeur laser, suit l'objet en continu, en calculant les angles de "*pan/tilt*" en fonction d'un système "auditeur de mote de cricket" [Priyantha, 2005] porté par l'objet.

[Zhang et al., 2019] ont acquis des séquences vidéo avec une caméra de surveillance et ont reconnu une personne cible en faisant correspondre les informations fournies par le capteur inertiel du téléphone de la personne avec les caractéristiques de marche et d'azimut de cap extraites des vidéos. Ils ont appliqué une technique de suivi d'objet basée sur les réseaux de neurones convolutifs (CNN) afin de gérer l'occlusion.

[Bo Bo et al., 2015] a proposé une approche pour le suivi multi-personnes en temps réel dans des séquences vidéo de caméras statiques. La méthode proposée est basée sur un algorithme glouton de maximisation de la vraisemblance qui traite les images vidéo en temps réel et estime les trajectoires de plusieurs individus dans la scène. L'algorithme fonctionne en détectant d'abord les personnes dans les images vidéo à l'aide d'un réseau neuronal profond. Les détections sont ensuite liées entre les images à l'aide d'un algorithme glouton qui maximise la probabilité des pistes en fonction de l'apparence et des caractéristiques de mouvement des individus. L'algorithme intègre également un terme de régularisation de trajectoire pour améliorer la cohérence des trajectoires dans le temps.

Il est important de noter que les solutions déployant des caméras statiques sont rarement utilisées seules sans être fusionnées à d'autres informations. [Dong et al., 2021] a proposé une approche qui fusionne les caméras statiques et mobiles. Parmi les caméras statiques, seules quelques-unes sont calibrées au départ. À chaque fois qu'un piéton est localisé dans le champ de vision d'une caméra calibrée et qu'il rentre dans le champ de vision d'une caméra non

calibrée, cette dernière est calibrée automatiquement via *Perspective-n-Point* [Lu, 2018]. Leur système atteint une précision de 0,68 m en moyenne.

[Zhang et al., 2022] a proposé une méthode de localisation en deux étapes pour les environnements intérieurs qui fusionne les données de plusieurs capteurs pour obtenir une grande précision. Dans la première étape, une localisation grossière est effectuée à l'aide d'empreintes WiFi fixes et de localisation par caméras fixes. Dans un second temps, une localisation fine est réalisée à l'aide d'une IMU pour accomplir une PDR. Un pistage visuel est mis en place en combinant *YOLOv5* [Jocher et al., 2020] et *DeepSORT* [Wojke et al., 2017]. Les auteurs présentent des résultats expérimentaux montrant que la méthode proposée atteint une précision moyenne de 5 cm. La méthode a des applications potentielles dans la navigation intérieure, le suivi des agents et les services basés sur la localisation. L'article souligne l'importance de combiner plusieurs modalités de capteurs avec les caméras pour la localisation intérieure.

6.2 Caméra mobile

L'approche analogue est de déployer une caméra attachée au piéton. Ensuite, le déplacement est déduit à partir du changement dans l'image acquise. Ces approches peuvent être catégorisées en fonction du type de caméra utilisée, du type d'amers détectés dans l'environnement intérieur et du traitement des données par un algorithme traditionnel ou une intelligence artificielle.

6.2.1 Caméras 2D vs 3D

[Gang and Pyun, 2019] a proposé une approche de positionnement intérieur utilisant une technologie de localisation hybride qui combine caméra 2D, Bluetooth et les données des capteurs inertiels et magnétiques. Le système proposé utilise les capteurs intégrés et les capacités de communication sans fil des téléphones pour estimer l'emplacement de l'utilisateur dans un environnement intérieur. Le système utilise des signaux Bluetooth pour détecter la proximité de l'utilisateur avec des points de référence, tels que des points d'accès et des balises, et intègre des données de capteur, telles que des mesures d'accéléromètre et de gyroscope, pour estimer le mouvement et l'orientation de l'utilisateur. La caméra 2D sert à détecter des QR codes pré-positionnés dans le bâtiment. La distance à ces QR codes est estimée en connaissant la position de celui-ci ainsi que sa taille ce qui permet d'appliquer une mesure milliradian.

[Lu et al., 2017] a proposé une méthode de localisation en intérieur basée sur des images et des vidéos multi-vues. La méthode proposée combine l'odométrie visuelle et la structure à partir du mouvement pour estimer la trajectoire 3D d'un utilisateur et le localiser dans un environnement intérieur. Cette méthode utilise plusieurs caméras pour capturer des images et des vidéos de l'environnement, puis extrait les caractéristiques de ces sources de données pour créer des nuages de points 3D. Les nuages de points sont ensuite utilisés pour estimer la trajectoire de l'utilisateur en suivant son mouvement entre les images.

[Akal et al., 2018] a proposé une approche pour la localisation basée sur l'image à l'aide d'un système de détection distribué. L'approche proposée utilise plusieurs caméras montées sur une seule plate-forme pour capturer des images de l'environnement sous différentes perspectives. Le système fonctionne en générant d'abord un modèle 3D de l'environnement à l'aide des images capturées par les caméras. Le modèle 3D est ensuite utilisé pour estimer la position et l'orientation de l'utilisateur sur la base des images capturées par une seule caméra.

Le processus d'estimation prend en compte la position relative et l'orientation des caméras pour améliorer la précision de la localisation. Les résultats expérimentaux pour différentes expériences présentent une erreur de localisation moyenne de l'ordre de 15 cm.

6.2.2 Amers artificiels vs organiques

[Benligiray et al., 2019] a proposé un système de marqueurs appelé *STag*, qui est conçu pour être stable et robuste dans une variété de conditions. Les marqueurs fiduciaires sont utilisés dans les applications de vision par ordinateur pour fournir un point de référence pour le suivi et la localisation d'objets. Le système *STag* est basé sur un ensemble de marqueurs circulaires disposés selon un motif unique, qui peuvent être détectés et reconnus par des algorithmes de vision par ordinateur. Ces marqueurs sont conçus pour être reconnaissables et stables même dans des conditions difficiles telles que faible luminosité, occlusion partielle et flou de mouvement. L'article met en avant la robustesse et stabilité de l'emploi des marqueurs fiduciaires. Les auteurs comparent également le système *STag* avec d'autres systèmes de marqueurs *AR-Toolkit* [ARToolKit, 2004], *ArUco* [Garrido-Jurado et al., 2014] et *RUNE-Tag* [Bergamasco et al., 2016]. Ils montrent qu'il les surpasse en termes de stabilité et de robustesse.

[Ooi et al., 2018] a proposé une approche pour localiser les nœuds de capteurs mobiles sur des robots 4 roues à l'aide de QR codes et par odométrie avec correction d'erreurs. L'approche proposée utilise des QR codes, qui sont placés à des emplacements connus dans l'environnement, comme points de référence pour la localisation. Les nœuds de capteurs mobiles utilisent une caméra pour détecter les codes QR et estimer leur position et leur orientation dans l'environnement. Le système intègre également les données d'odométrie pour améliorer la précision de la localisation. Dans ce cas, l'amer choisi est totalement artificiel.

[Xiao et al., 2018] a proposé un système de positionnement intérieur qui utilise des objets statiques dans de grandes scènes intérieures et des caméras de téléphone. Le système proposé identifie et utilise des objets statiques tels que des portes, des fenêtres et des piliers comme points de repère pour estimer la position de l'utilisateur. Il fonctionne en construisant d'abord un modèle 3D de l'environnement intérieur à l'aide d'images capturées par les caméras des téléphones. Le modèle 3D est ensuite utilisé pour détecter et identifier des objets statiques pouvant servir de points de repère. Une fois les points de repère identifiés, le système les utilise pour estimer la position de l'utilisateur en fonction de leur distance et de leur angle par rapport aux points de repère. L'article présente des résultats expérimentaux qui présente une précision moyenne de 0,7 m, même dans des scénarios difficiles tels que des conditions de faible luminosité et des occlusions.

[Árvai and Homolya, 2019] a proposé une approche de localisation intérieure à l'aide de cartes radio Wi-Fi. il déploie un processus en deux étapes consistant en un filtrage et une correspondance d'empreintes de signaux. Dans l'étape de filtrage, le système utilise un modèle probabiliste pour filtrer les points d'accès non pertinents des signaux Wi-Fi reçus. Lors de l'étape de correspondance des empreintes, le système compare les signaux Wi-Fi filtrés avec une carte radio prédéfinie pour estimer la position de l'utilisateur. L'amer est donc organique car il est naturellement présents dans le bâtiment même s'il faut pré-établir la base de données reliée à celui-ci. Même si les amers artificiels permettent d'avoir une précision plus élevée aux systèmes qui les utilisent, ils peuvent être jugés intrusifs, trop coûteux ou impossibles à déployer [Zhu et al., 2019]. Dans ce cas l'amer organique est préféré.

6.2.3 Localisation et cartographie simultanées (SLAM)

SLAM est une méthode qui combine de manière itérative deux étapes distinctes : la localisation et la cartographie. Pour se localiser, un SLAM utilise une estimation de la position, puis construit une cartographie de l'environnement à partir des caractéristiques relevées dans les signaux collectés. L'objectif est de raffiner la position et la cartographie en combinant les deux étapes de manière itérative. L'interdépendance de ces deux étapes est la problématique à laquelle le SLAM veut répondre.

Dans son instantiation, le SLAM se découpe en deux blocs principaux *front-end* et *back-end*. Pour la majorité, l'objectif principal du bloc *front-end* est d'estimer la pose du véhicule donnée par sa position et son orientation. Donc, ce bloc s'occupe de l'acquisition et du traitement des données afin d'en extraire une première donnée estimation qui est fourni au second. À cause de facteurs externes, il y a inévitablement des erreurs d'estimation qui ne font qu'augmenter avec le temps [Placed et al., 2023]. Le *back-end* présente une représentation abstraite des données pour créer l'environnement du modèle. C'est donc ce bloc qui a pour but de limiter l'accumulation des erreurs d'estimation du *front-end* via des algorithmes de filtrage ou d'optimisation tels que la fermeture de boucle ou l'ajustement de faisceaux [Placed et al., 2023, Chghaf et al., 2022]. Ce bloc possède une contre réaction vers le *front-end* afin de corriger les erreurs d'estimation.

Il existe beaucoup de variantes du SLAM qu'il soit uniquement visuel [Macario Barros et al., 2022], visuel-inertiel [Servières et al., 2021] ou multimodal [Chghaf et al., 2022]. L'étude du SLAM ainsi que la classification de ces variantes est en dehors de ce manuscrit. Néanmoins, nous nous intéressons aux applications du SLAM dans le cas du piéton en espace intérieur et aux modalités exploitées par celui-ci.

ORB-SLAM est l'algorithme visuel monoculaire le plus connu. il a été proposé par [Mur-Artal et al., 2015]. C'est un système qui utilise les caractéristiques *ORB* [Ruble et al., 2011] pour le suivi et la cartographie. Il peut être initialisé avec des estimations de mouvement pour améliorer le suivi. Le système est également conçu pour être robuste aux fermetures de boucles et est évalué sur divers ensembles de données pour montrer des précisions moyenne entre 2 et 20 m sur différentes séquence du jeu de données *KITTI* [Geiger et al., 2013]. L'article démontre qu'*ORB-SLAM* atteint une précision et une robustesse élevées dans divers environnements et peut fonctionner en temps réel sur différentes plates-formes telles que les ordinateurs portables et les téléphones.

ORB-SLAM3 [Campos et al., 2020] est construit sur [Mur-Artal et al., 2015] et [Mur-Artal and Tardós, 2017]. Le système est conçu pour fonctionner dans plusieurs modes, y compris monoculaire, stéréo, inertiel visuel et cartes multiples. *ORB-SLAM3* exploite une variété de techniques, y compris les caractéristiques *ORB*, la détection de fermeture de boucle et l'optimisation de la carte, pour obtenir une précision et une robustesse élevées dans différents environnements. L'article évalue les performances d'*ORB-SLAM3* sur divers ensembles de données, montrant qu'il surpasse les versions précédentes d'*ORB-SLAM* et d'autres systèmes SLAM de pointe. Par exemple, dans sa version utilisant la camera monoculaire et les données inertielles, il présente une erreur moyenne de 0,114 m dans les couloirs, de 1,405 m dans les séquences de marches de longueur (jusqu'à 900 m), de 27,23 m en extérieur, de 0.0183 m en salle et de 0,9 m dans les séquences toboggans (privé de lumière et mouvement rapide).

RGB-D SLAM [Endres et al., 2013] présente une méthode pour construire des cartes 3D d'environnements intérieurs à l'aide d'une caméra RGB-D, telle que *Microsoft Kinect*. L'article décrit le processus d'acquisition et de traitement d'images de profondeur et de couleur à partir

de la caméra pour créer une représentation en nuage de points 3D dense de l'environnement. Les auteurs présentent également une méthode pour améliorer la précision du nuage de points en utilisant des techniques d'estimation de normale de surface et d'ajustement de plan. L'article évalue la méthode proposée sur plusieurs ensembles de données et la compare avec d'autres méthodes de cartographie 3D de pointe. Les résultats montrent que la méthode proposée atteint une précision moyenne de 0,260 m ; 0,087 m ; 0,057 m et 0,860 m pour 4 scénarios différents.

CNN-SLAM [Tateno et al., 2017] présente une méthode de SLAM monoculaire dense en temps réel utilisant un réseau neuronal convolutif (CNN) pour la prédiction de profondeur. Les auteurs proposent un système qui estime conjointement la pose de la caméra et les cartes de profondeur dense en temps réel, en utilisant un CNN pour prédire les profondeurs à partir d'une seule image monoculaire. Le système proposé utilise une approche directe pour optimiser l'erreur photométrique entre les intensités d'image prédites et mesurées. L'article évalue la méthode proposée sur plusieurs ensembles de données difficiles et montre qu'elle surpasse les méthodes SLAM monoculaires de pointe existantes en termes de précision et de robustesse. La méthode proposée permet également d'obtenir des performances en temps réel sur un ordinateur de bureau standard. Dans l'ensemble, l'article démontre le potentiel des méthodes d'apprentissage en profondeur pour le SLAM et montre que les méthodes basées sur CNN peuvent atteindre une précision élevée et des performances en temps réel pour le SLAM monoculaire.

L'usage de modèles basées sur l'IA dans le SLAM importe un nouveau type d'information dite sémantique où les amers et caractéristiques détectées sont étiquetées afin de permettre une nouvelle recherche de correspondance entre une détection et la carte construite. [Zhang et al., 2019] présente une méthode pour améliorer l'association d'objets dans le SLAM sémantique, qui est obtenue en utilisant des modèles thématiques hiérarchiques (*hierarchical topic models*) qui intègrent à la fois des informations contextuelles locales et globales sur les objets. L'approche améliore la précision de l'association d'objets en tenant compte à la fois des informations visuelles et textuelles des objets détectés dans la scène. L'approche proposée peut être facilement intégrée dans les systèmes SLAM sémantiques existants. [Wang et al., 2019] propose une solution pour le SLAM sémantique dans des scènes dynamiques, où les objets peuvent se déplacer et changer d'apparence au fil du temps. L'approche utilise une combinaison d'un réseau de détection d'objets pré-entraîné et d'un algorithme de suivi personnalisé pour détecter et suivre des objets en temps réel. L'algorithme de suivi utilise une contrainte de cohérence temporelle pour améliorer la précision du suivi de l'objet tout en minimisant le coût de calcul.

[Nicholson et al., 2018] présente une méthode appelée *QuadricSLAM* qui utilise des contraintes quadriques duales extraites de détections d'objets comme amers dans le SLAM sémantique. Le procédé utilise un réseau de détection d'objets pré-entraîné pour détecter des objets dans l'environnement et génère des quadriques doubles pour représenter leur forme et leur position en 3D. Ces quadriques sont ensuite utilisées comme points de repère dans SLAM pour améliorer la précision de l'estimation de la trajectoire de la caméra et de la reconstruction de la carte.

[Yu et al., 2019] présente un modèle d'observation qui améliore la précision de la détection d'objets 3D pour les systèmes SLAM probabilistes. Le modèle proposé est basé sur un cadre probabiliste qui prend en compte les variations de taille, de forme et d'orientation des objets. Il estime la pose de l'objet en calculant son centre et son orientation. Le modèle proposé utilise la

matrice de covariance pour représenter les variations de la forme et de l'orientation de l'objet. L'efficacité du modèle est améliorée en utilisant l'approche variationnelle pour l'optimisation. La précision du modèle proposé est évaluée en fonction du taux de détection d'objet, de l'erreur de détection d'objet et de l'exhaustivité du modèle d'objet. Les résultats montrent que le modèle proposé surpasse les méthodes de pointe en termes de précision et d'efficacité de calcul.

En 2020, *Kimera SLAM* est proposé par [Rosinol et al., 2020] dont le fonctionnement est résumé dans la figure 6.1. *Kimera SLAM* est une bibliothèque open-source qui fournit des capacités de localisation et de mappage métriques-sémantiques en temps réel. *Kimera* utilise des caméras monoculaires, stéréo ou RVB-D pour capturer les données de l'environnement et traite les données pour créer une carte 3D de l'environnement. La bibliothèque intègre également des informations sémantiques dans le processus de cartographie, permettant la création de cartes qui peuvent différencier différents types d'objets, tels que des tables, des chaises et des murs. *Kimera* est conçue pour fonctionner avec une variété de capteurs, ce qui en fait un outil polyvalent pour les chercheurs et les développeurs travaillant sur des applications robotiques. L'article traite des détails techniques de la bibliothèque *Kimera*, y compris son utilisation de graphes de pose pour représenter la trajectoire du robot et sa cartographie de l'environnement. Il présente également plusieurs expériences démontrant l'efficacité de *Kimera* pour des tâches telles que la détection de fermeture de boucle et le suivi dynamique d'objets. Dans l'ensemble, l'article présente *Kimera* comme un outil puissant pour la localisation et la cartographie métrique-sémantique en temps réel, avec des applications potentielles dans divers domaines, notamment la robotique, la réalité augmentée et les véhicules autonomes.

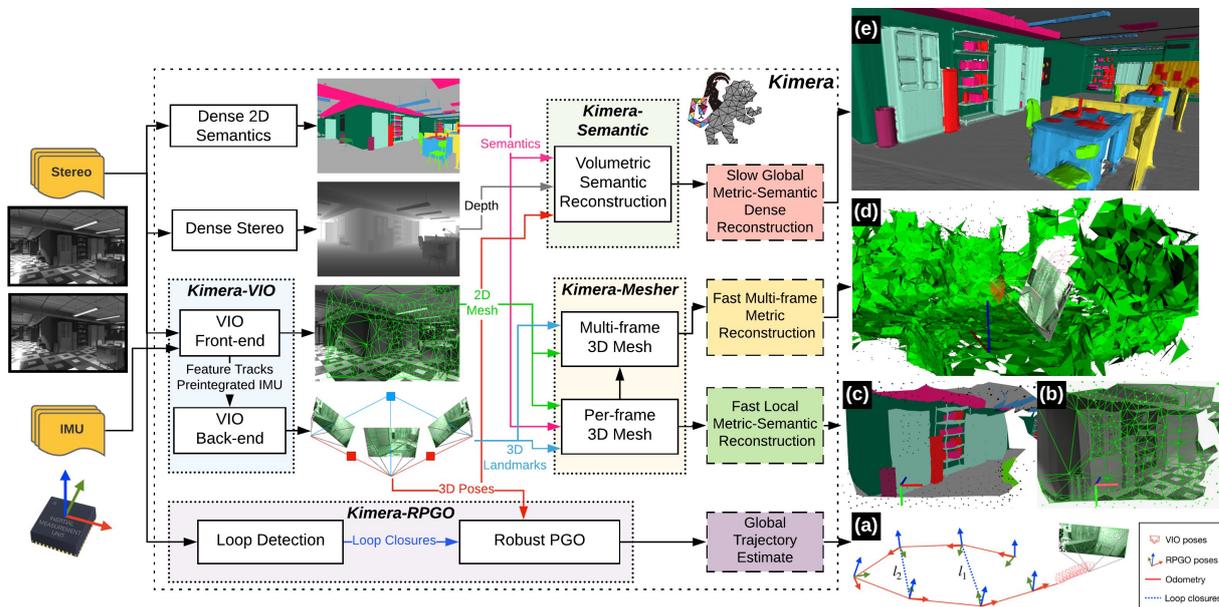


FIGURE 6.1 – Schéma récapitulatif de *Kimera* de [Rosinol et al., 2020].

[Li et al., 2022a] propose une méthode basée sur la fusion multi-capteurs pour construire une carte sémantique d'un environnement intérieur avec une capacité de localisation à piste unique. La méthode proposée utilise des données de capteurs LiDAR, RGB-D et des capteurs inertiels pour extraire des informations sémantiques et construire une carte qui peut

être utilisée pour la localisation. Les informations sémantiques sont extraites à l'aide d'une combinaison de techniques de détection d'objets et d'extraction de caractéristiques basées sur l'apprentissage profond. La localisation est effectuée à l'aide d'un UKF qui fusionne les données des différents capteurs pour estimer la position du robot en temps réel. [Li et al., 2022b] présente une approche SLAM qui utilise le *crowdsourcing* pour collecter des données de capteurs et construire des cartes sémantiques. Les données des capteurs sont collectées à l'aide de smartphones et d'appareils portables transportés par la foule. Les cartes sont construites en fusionnant les données du capteur à l'aide d'une technique d'optimisation basée sur des graphiques. La méthode proposée est capable de construire des cartes de haute qualité qui peuvent être utilisées à des fins de localisation.

6.2.4 Réalités augmentée (AR), virtuelle (VR) et mixte (MR)

La réalité augmentée (AR) est une technologie qui ajoute des éléments numériques à l'environnement réel de l'utilisateur en temps réel, souvent en utilisant un appareil mobile ou des lunettes spéciales. Les informations numériques sont superposées sur le monde réel, permettant ainsi à l'utilisateur de voir et d'interagir avec des objets numériques tout en restant connecté à son environnement physique. La réalité virtuelle (VR) est une technologie qui permet à un utilisateur de s'immerger dans un environnement numérique simulé en utilisant un casque. Cette technologie crée une expérience sensorielle immersive en simulant la vue, le son et le toucher pour donner l'impression à l'utilisateur qu'il est physiquement présent dans un monde virtuel. La réalité mixte (MR) est une technologie qui combine des éléments de la VR et de la AR. La MR permet aux utilisateurs de voir et d'interagir avec des objets numériques tout en restant connectés à leur environnement physique, mais à la différence de la AR, les objets numériques sont placés dans l'environnement réel de l'utilisateur comme s'ils étaient réellement présents. Cela permet aux utilisateurs d'interagir avec des objets numériques de manière plus réaliste et naturelle qu'avec la AR. Ces trois technologies ont chacune besoin de la position et de l'orientation de l'utilisateur ainsi que d'une acquisition d'images. Elles se placent donc naturellement comme des solutions de localisation piétonnes.

[Yan et al., 2015] a proposé une approche de la navigation intérieure à l'aide d'un système mobile de réalité augmentée. Le système proposé utilise un appareil mobile, tel qu'un smartphone ou une tablette, équipé d'une caméra et de capteurs inertiels, pour offrir aux utilisateurs une expérience de navigation interactive et immersive. Le système superpose des informations virtuelles, telles que des directions et des points d'intérêt, sur le flux vidéo en direct capturé par la caméra, et fournit un retour audio et visuel pour guider l'utilisateur vers sa destination.

[Gerstweiler, 2018] présente un système de AR qui peut guider les personnes dans des environnements intérieurs complexes. Le système utilise une caméra de profondeur et un algorithme SLAM pour créer une carte 3D de l'environnement et suivre la position et l'orientation de l'utilisateur en temps réel. Le système utilise également des techniques d'apprentissage automatique pour reconnaître les objets et fournir des informations contextuelles à leur sujet à l'utilisateur. Le système est évalué dans un environnement réel et les résultats démontrent la faisabilité de l'utilisation de la technologie AR pour la navigation et le guidage en intérieur.

[Baek et al., 2019] propose un système de localisation intérieure basé sur l'image utilisant une approche de AR pour la gestion des installations. Le système utilise une base de données d'images panoramiques, ainsi que des caractéristiques telles que les coins et les bords, pour reconnaître et localiser les positions des utilisateurs dans les environnements intérieurs. Les

auteurs ont mis en œuvre leur approche sur un appareil mobile et ont montré sa précision et sa faisabilité dans un bâtiment du monde réel. Ils ont également créé un prototype d'application pour les tâches de gestion des installations/usines, qui permet aux utilisateurs de recevoir des conseils visuels et des informations sur les objets dans leur environnement. Le système proposé a le potentiel d'améliorer l'efficacité de la gestion des installations/usines en fournissant des conseils en temps réel aux utilisateurs, en améliorant leur connaissance de la situation et en réduisant les erreurs.

[Schenkluhn et al., 2023] propose un système de positionnement intérieur basé sur la AR pour la domotique intelligente. Le système utilise les capteurs intégrés d'un smartphone pour suivre la position et l'orientation de l'utilisateur en temps réel. Le système proposé utilise un plan d'étage de la maison comme référence pour estimer l'emplacement de l'utilisateur, et une superposition AR pour fournir des repères visuels pour guider l'utilisateur vers l'emplacement cible. Le système est conçu pour contrôler les appareils domestiques intelligents tels que les lumières, les thermostats et les systèmes de sécurité, et est évalué sur un prototype à petite échelle d'une maison intelligente. Les résultats montrent que le système proposé est efficace pour guider les utilisateurs vers leurs destinations souhaitées dans la maison intelligente.

6.2.5 Traitement traditionnel vs intelligence artificielle

La localisation par traitement d'image voit aussi l'intégration de modèles d'IA contrairement aux modèles traditionnels qui ne sont pas basés sur l'apprentissage. L'IA comprend beaucoup d'algorithmes de localisation pour les systèmes qui utilisent les appareils photo mobiles.

[Bai et al., 2014] a construit une base de données de points de repère en utilisant un télémètre laser pour mesurer la distance entre l'emplacement de la caméra et les points de repère sélectionnés. Durant la localisation, les images acquises avec la caméra mobile sont comparées avec celles de la base de données à l'aide d'algorithmes d'appariement de caractéristiques tels que *SIFT* [Lowe, 2004], *SURF* [Bay et al., 2006] ou *ORB* [Rublee et al., 2011]. Le temps de traitement de cette étape est fortement affecté par le nombre d'images/de caractéristiques dans la base de données qui doivent être extraites par rapport aux images du flux vidéo de la caméra mobile [Morar et al., 2020].

[Marouane et al., 2014] a proposé une approche de systèmes de positionnement visuel en étendant l'architecture *MoVIPS* (*Mobile Visual Information Provider for Navigation Services*). L'approche proposée utilise une combinaison de capteurs visuels, tels que des caméras, et des technologies de communication sans fil, telles que Wi-Fi et Bluetooth, pour fournir des informations de positionnement précises et fiables dans des environnements intérieurs. Le système utilise des techniques d'extraction et d'appariement de caractéristiques visuelles pour identifier et suivre des points clés dans l'environnement, et intègre les informations visuelles avec les signaux sans fil pour estimer la position de l'utilisateur.

[Huang et al., 2018] a proposé une approche du positionnement intérieur utilisant des caractéristiques de l'environnement 3D. Il utilise une méthode d'extraction de caractéristiques 3D qui exploite la caméra et le capteur de profondeur d'un appareil mobile pour générer un modèle 3D de l'environnement. Le système extrait les caractéristiques 3D du modèle et les utilise pour estimer la position de l'utilisateur en temps réel. L'approche est conçue pour surmonter les limitations des méthodes basées en 2D, qui peuvent être sujettes à des erreurs dues aux occlusions et aux changements des conditions d'éclairage.

[Lu et al., 2017] a proposé un modèle de régression multi-vues pour déterminer la localisation et l'orientation de l'utilisateur avec précision. [Xiao et al., 2018] a utilisé le réseau de neurones Faster-RCNN [Ren et al., 2015] pour la détection et l'identification d'objets statiques dans les images acquises avec les caméras du smartphone. YOLO [Redmon et al., 2016] est aussi utilisé pour cette tâche. Un autre CNN profond, *Convnet*, a été utilisé dans le système de localisation proposé par [Akal et al., 2018]. Il propose d'effectuer la localisation en se basant sur les images à l'aide d'un système de détection distribué. L'approche proposée utilise plusieurs caméras placées à différents endroits dans l'environnement pour capturer des images de la scène à partir de différents points de vue. Le système utilise ensuite un algorithme de détection distribué pour estimer la position de l'utilisateur en combinant les différentes informations. L'approche est conçue pour surmonter les limites des méthodes traditionnelles, qui peuvent souffrir d'occlusion, d'une faible visibilité ou d'une couverture limitée. Ce réseau utilise des composés images de quatre images monoculaires non superposées placées sur un robot au sol, atteignant une précision centimétrique, mais nécessitant un ensemble de données important d'images composées pour l'entraînement.

Les solutions basées sur l'apprentissage automatique ont atteint des temps de calcul interactifs ou même en temps réel. Les systèmes utilisant ces modèles peuvent atteindre une précision de localisation de moins d'un mètre à des dizaines de centimètres [Morar et al., 2020]. Ces solutions semblent avoir de meilleures performances de précision par rapport aux solutions traditionnelles, tandis qu'elles bénéficient des mêmes avantages de ne pas nécessiter de déploiement de repères visuels artificiels dans l'espace intérieur. Donc, pour notre besoin de générer un signal, nous souhaitons détecter un objet qui existe dans l'espace en intérieur. Dans ce cas, nous nous intéressons aux détecteur d'objets à partir d'une image faisant appel à un modèle d'intelligence artificielle. En particulier, nous nous intéressons à YOLOv5 [Jocher et al., 2020] pour les raisons qui sont mentionnés ci-dessous.

6.3 Détecteurs d'objets dans une image

6.3.1 YOLO

You Only Look Once (YOLO) est une approche de détection d'objets dans des images en temps réel basée sur un Réseau de Neurones Convolutionnel (CNN) entraîné par un apprentissage supervisé. Contrairement aux méthodes traditionnelles qui impliquent de passer l'image à travers un modèle de détection plusieurs fois à différentes échelles, YOLO adopte une approche différente en traitant le problème de manière globale. Au lieu de diviser l'image en régions et d'effectuer des prédictions séparées, YOLO divise l'image en une grille et prédit les boîtes englobantes et les classes d'objets directement à partir de cette grille. D'où le fait qu'il est appelé "regarde une seule fois".

Il existe plusieurs versions de YOLO qui sont des améliorations incrémentales les unes des autres ou des variantes du modèle principal faites dans le but de répondre à une tâche. Le premier modèle YOLOv1 est proposé par [Redmon et al., 2016] qui est amélioré en YOLOv2 [Redmon and Farhadi, 2017] et ensuite en YOLOv3 [Redmon and Farhadi, 2018]. En avril 2020, YOLOv4 est proposé par [Bochkovskiy et al., 2020]. Enfin, en juin 2020, l'entreprise *Ultralytics* publie YOLOv5 [Jocher et al., 2020] sous la forme d'un logiciel libre. Depuis, YOLO est en évolution continue sur cette version faisant de YOLOv5 la référence. Un historique détaillé de YOLO ainsi que les différences principales sont publiés dans [Jiang et al., 2022].

Dans la figure 6.2, le diagramme de l'architecture de YOLO résume le fonctionnement de celui ci. La version détaillée en blocs élémentaires est présentée à l'annexe C.3. YOLO possède 25 couches qui sont découpées en 4 grande parties. La partie "*Backbone*" est une série de couches de neurones convolutifs qui ont pour tâche d'extraire des caractéristiques bas niveau de l'image. La partie "*Neck*" est chargée de la construction des pyramides de caractéristiques i.e. elle arrange les caractéristiques extraites pour former des ensembles hiérarchisés qui sont les caractéristiques haut niveau d'un objet.

La partie "*Head*" est une série de couches totalement connectées ("*Fully connected*"). C'est cette partie qui va apprendre à combiner des caractéristiques provenant des pyramides de "*Neck*" pour effectuer une détection dite "multi-échelle". En utilisant des couches de regroupement (*pooling*) et des convolutions à différentes échelles, YOLO est capable de détecter des objets de différentes tailles dans l'image. Il propose donc trois échelles : grand, moyen et petit où les objets plus petits sont plus difficiles à détecter. Cela permet au modèle d'être robuste face à des objets de tailles variées et d'obtenir de bonnes performances sur des ensembles de données diversifiés.

À la sortie de "*Head*", deux éléments sont produits : un ensemble de plusieurs candidats est proposé sous la forme de boîtes englobantes autour de la détection potentielle ; et une carte de probabilité de classes qui indique la probabilité pour qu'une cellule de la grille appartienne à une classe d'objet.

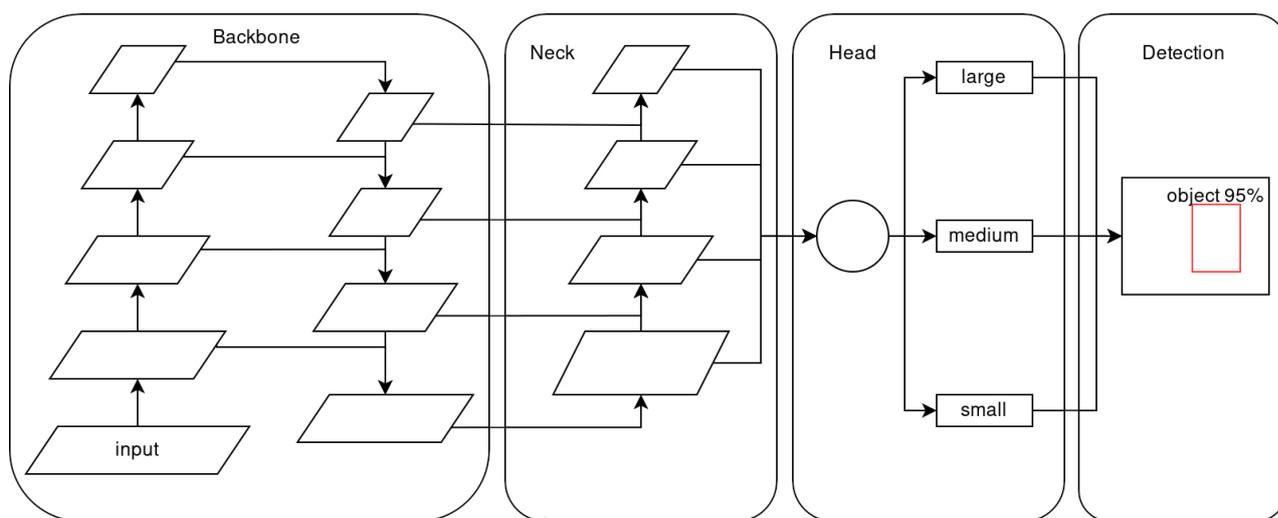


FIGURE 6.2 – Diagramme récapitulatif de YOLOv5. Image adaptée de [Azam et al., 2022].

L'objet mathématique fondamental dans YOLO est la boîte englobante. Pour chaque image, YOLO génère une grille (par exemple de taille $S \times S$). Chaque cellule de la grille est responsable de la prédiction de plusieurs boîtes englobantes, et des scores de confiance pour ces boîtes. Chaque boîte englobante est définie par cinq composantes : $(x, y, w, h, \text{score de confiance})$, où :

- (x, y) désigne le centre de la boîte englobante par rapport aux limites de la cellule de la grille.
- (w, h) sont la largeur et la hauteur de la boîte, par rapport à l'image.
- Le score de confiance représente la probabilité que la boîte contienne un objet et dans quelle mesure la boîte s'adapte à l'objet.

Chaque cellule de la grille prédit également les probabilités de classe C (une par classe pour toutes les C classes). Ces probabilités sont indépendantes des boîtes englobantes. L'étape "Detection" donne la prédiction finale de YOLO pour chaque boîte englobante qui est donnée par le produit du score de confiance et des probabilités de classe en éliminant les non maxima parmi toutes les propositions.

La fonction de perte de YOLO consiste en une perte de régression (pour les coordonnées de la boîte englobante), une perte de classification (pour les prédictions de classe) et une perte de confiance (pour l'objectivité, i.e. à quel point la boîte englobante est cohérente avec l'objet). Il est conçu de manière à équilibrer le compromis entre la localisation de l'objet (trouver la bonne grille et la boîte englobante) et sa classification correcte.

Cela permet à YOLO de réaliser la détection en temps réel, même sur des plates-formes avec des ressources limitées. De plus, YOLO est capable de détecter plusieurs objets simultanément, ce qui en fait un choix idéal pour les applications nécessitant une détection multi-cibles où le compromis entre rapidité et précision est nécessaire.

YOLO utilise en entrée une résolution d'image fixe (px \times px) : soit 640 \times 640, soit 1280 \times 1280. Cette première spécification rend la division de l'image en grille consistante et est raisonnable en considérant que la résolution d'une caméra n'est généralement pas dynamique.

L'autre avantage majeur de YOLO est qu'il est proposé en plusieurs tailles de modèle qui sont listés dans le tableau 6.1 (de nano, à XL). Ils sont comparés en terme de mAP qui est une métrique d'évaluation présentée en section 7.4, en vitesse d'exécution sur CPU et sur GPU, en nombre de paramètres et en intensité arithmétique exprimée en FLOPs. Chaque modèle présente un compromis entre tous ces facteurs où un modèle plus petit est plus rapide avec un impact moindre sur la mémoire mais sera moins précis dans ses détections.

Model	Résolution carrée (pixels)	mAPval 50-95	mAPval 50	Speed CPU (ms)	Speed GPU (ms)	Params (10 ⁶)	FLOPs (10 ⁹)
YOLOv5n	640	28.0	45.7	45	6.3	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	140.7	209.8

TABLE 6.1 – Tableau de comparaison en précision et temps d'exécution [Jocher et al., 2020].

La figure 6.3 donne une visualisation de la comparaison du tableau 6.1. Chaque courbe implémente une variation du nombre des points d'ancrage pour le positionnement des boîtes englobantes. Plus le nombre de points d'ancrage est élevé, plus le nombre de boîtes englobantes candidates est élevé, plus le temps d'inférence d'une version de YOLO est important. Dans les chiffres présentés, le temps d'exécution GPU est calculé en inférant YOLO sur *AWS p3.2xlarge V100*¹. YOLO y est comparé sur le jeu de données *COCO2017val* [Lin et al., 2014]

1. <https://aws.amazon.com/ec2/instance-types/p3/>

à *EfficientDet* [Tan et al., 2020] de *Google Research, Brain Team* qui est considéré comme référence.

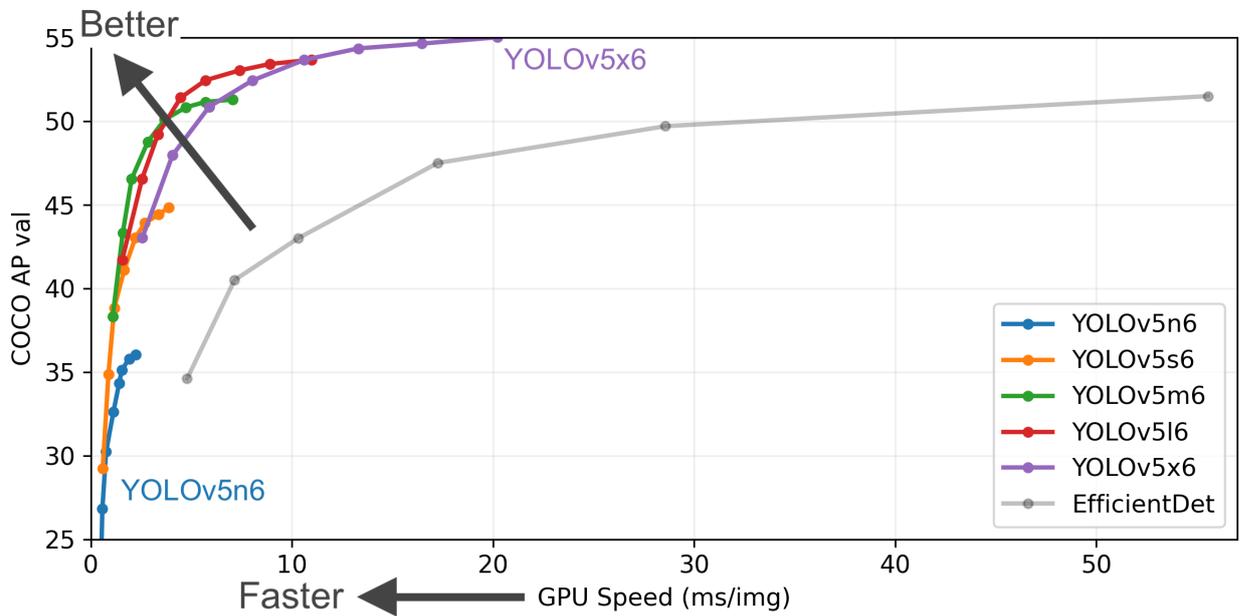


FIGURE 6.3 – Résultats d'évaluation des différentes tailles de YOLO par [Jocher et al., 2020].

La famille YOLO a eu un impact considérable dans de nombreuses applications de la vision par ordinateur. Elle a été largement adoptée dans des applications telles que la navigation en intérieur [Wang et al., 2022], la surveillance vidéo [Narejo et al., 2021], l'imagerie satellite [Li et al., 2020], la détection d'objets pour les robots [Souidene Mseddi et al., 2021, Mingyuan et al., 2019], l'agriculture [Yao et al., 2021] et bien d'autres [Neethirajan, 2022].

6.3.2 Concurrents de YOLO

"Regions with Convolutional Neural Networks" (RCNN) est un modèle de détection d'objets qui a été proposé par [Girshick et al., 2014]. Dans le modèle RCNN, une recherche sélective est d'abord effectuée pour identifier les régions d'intérêt potentielles, suivie par la mise en œuvre de CNN pour classer ces régions. Les boîtes englobantes sont ensuite affinées à l'aide d'une régression. Bien que RCNN soit précis, il est lent en raison de l'extraction des fonctionnalités pour chaque proposition de région, ce qui a conduit à l'émergence de Fast RCNN et Faster RCNN.

Fast RCNN de [Girshick, 2015] a été proposé pour résoudre le problème de performance du modèle RCNN. Contrairement à RCNN, *Fast* RCNN extrait les caractéristiques une fois pour chaque image et réalise ensuite une régression et une classification sur chaque région d'intérêt. Cela le rend nettement plus rapide que RCNN tout en conservant une précision similaire.

Faster RCNN de [Ren et al., 2015] propose un nouveau mécanisme de proposition de région appelé RPN (Region Proposal Network). Avec *Faster* RCNN, la recherche sélective est remplacée par un RPN entièrement différentiable, rendant le processus de proposition de région aussi rapide que l'extraction de fonctionnalités. *Faster* RCNN peut donc être entraîné en une seule étape, ce qui améliore à la fois la rapidité et la précision.

Mask RCNN de [He et al., 2017] ajoute une couche supplémentaire au *Faster* RCNN pour la segmentation d'objets, ce qui lui permet non seulement de détecter des objets mais aussi de délimiter précisément leurs contours en attribuant chaque pixel à une classe. Cela se fait grâce à l'ajout d'un réseau de masques parallèle au réseau de classification et de boîtes englobantes existant. *Mask* RCNN est donc capable de fournir des résultats de détection et de segmentation d'objets de haute qualité mais n'est pas plus rapide que *Faster* RCNN.

Les Transformers sont des modèles de traitement du langage naturel originellement proposés par [Vaswani et al., 2017]. Ils ont prouvé leur efficacité dans les tâches de traitement du langage naturel, notamment ChatGPT-4 [OpenAI, 2023], et commencent à être utilisés de plus en plus pour les tâches de vision par ordinateur. Leur principale innovation est le mécanisme d'attention, qui pèse l'importance relative de chaque données dans le contexte de la séquence. Cependant, l'application de l'architecture des Transformers à des images de grande taille peut être très coûteuse en termes de mémoire et de calcul.

Dans les Transformers, l'opération clé est l'attention auto-régressive, qui calcule une matrice d'attention de taille $N \times N$, où N est le nombre de "mots" dans une séquence ou de pixels dans l'image. Cela signifie que la complexité en espace (mémoire) et la complexité en temps (calcul) de l'opération de calcul de d'attention sont toutes deux quadratiques en fonction de la taille de l'image. Par conséquent, pour une image de 1000×1000 pixels, cela pourrait nécessiter de gérer une matrice d'attention de 1 million \times 1 million, ce qui est infaisable avec les ressources matérielles actuelles car il faudrait une mémoire d'au moins 10^{12} flottants pour contenir cette matrice dans un téléphone ce qui enfreint l'objectif [F] de ce manuscrit.

Pour résoudre ce problème, différentes approches ont été proposées, comme l'utilisation de "patches" d'images comme entrée pour le Transformer (comme dans le modèle Vision Transformer (ViT) [Dosovitskiy et al., 2020]) ou l'utilisation de mécanismes d'attention locale ou d'attention linéaire qui réduisent la complexité de l'opération d'attention [Zhu et al., 2020b]. Cependant, ces méthodes peuvent également avoir des compromis en termes de performance ou de précision de la détection d'objets.

Les différentes versions des familles YOLO, RCNN et Transformers sont comparées en terme de précision et vitesse d'exécution dans [Mekhali et al., 2021]. Dans [Li et al., 2020], YOLOv3 a dépassé en précision et en vitesse d'exécution *Faster* R-CNN pour la détection de parcelle agricole. Dans [Kim et al., 2020], YOLOv4 a dépassé en précision et en vitesse d'exécution *Faster* R-CNN pour la reconnaissance de véhicule. Enfin, les versions de YOLO ont été comparées entre elles par [Nepal and Eslamiat, 2022] et YOLOv5 montre de meilleurs résultats que les versions précédentes.

6.4 Synthèse

Les techniques présentées exploitent une ou plusieurs caméras. Quand ces caméras utilisées sont fixes, elles doivent déterminer la position du piéton par projection au sol. Les solutions déployant des caméras statiques sont rarement utilisées seules sans être fusionnées à d'autres informations. Afin d'augmenter la précision et de réduire les ambiguïtés, les données des caméras panoramiques fixes peuvent être fusionnées avec d'autres sources. Par exemple, il est possible de détecter les humains afin de les placer dans une carte de l'environnement intérieur. D'autres solutions ont utilisé des caméras fixes "pan/tilt" avec des réseaux de capteurs sans fil pour suivre des objets connectés. Certains ont utilisé des séquences vidéo pour reconnaître une personne cible en faisant correspondre les informations fournies par le capteur inertiel du

téléphone de la personne avec les caractéristiques de marche et d'azimut de cap extraites des vidéos. Pour effectuer un recalage visuel à partir d'images, les caméras statiques posent un inconvénient majeur : elles ne sont pas en communication direct avec l'appareil de localisation. Elles doivent donc communiquer avec le système de localisation les rendant assimilables aux systèmes communiquant vu en section 2.2.3 ce qui est incompatible avec l'objectif [G] de ce manuscrit.

Une approche proposée fusionne les caméras statiques et mobiles pour obtenir une grande précision. Cette méthode utilise *Perspective-n-Point* pour calibrer automatiquement les caméras non calibrées lorsqu'un piéton rentre dans leur champ de vision. Enfin, certaines méthodes fusionnent les données de plusieurs capteurs pour obtenir une grande précision en utilisant des empreintes WiFi et des caméras statiques pour effectuer une localisation grossière, puis utilise une IMU pour effectuer une localisation fine. Un pistage visuel est mis en place en combinant *YOLO* et *DeepSORT*. De manière similaire à l'exploitation du réseau de communication du bâtiment de la section 2.2.3, l'usage des caméras statiques requiert la présence de celles-ci dans l'espace intérieur. Il est nécessaire d'assurer la disponibilité de ces caméras ce qui n'est pas possible dans les situations d'urgences (incendie, coupure électrique, ...). De plus, la présence de caméras de surveillance pose des questions de confidentialité et libertaire et donc présente donc une réticence à être acceptée par le grand public montrant donc un désaccord avec l'objectif [G] de ce manuscrit.

Quand les caméras sont mobiles, elles peuvent être 2D ou 3D. Certaines approches utilisent des marqueurs artificiels tels que des QR codes ou des marqueurs de repère ("*fiducial markers*") pour fournir un point de référence pour la localisation. Les données de capteurs inertiels et magnétiques sont également utilisées pour estimer le mouvement et l'orientation de l'utilisateur et fusionnés aux données caméras. Différents systèmes ont été proposés pour estimer la trajectoire 3D de l'utilisateur qui se basent sur l'odométrie visuelle et la structure à partir du mouvement. Dans ce cas, les caméras produisent une information permettant le recalage visuel sur les amers détectés i.e cette information permet au système de modifier la position et/ou l'orientation du piéton en fonction de l'observation de l'amer détecté. Ce qui fait du recalage visuel un moyen d'introduire un signal extéroceptif dans un système proprioceptif à base d'événement de détection. Nous présentons notre mise en œuvre de ce recalage dans la section 8.1.

Les caméras peuvent exploiter la détection d'amers présents dans l'espace intérieur. Ces amers peuvent être artificiels ou organiques. Les premiers sont ajoutés dans le but d'effectuer la localisation et façonnés pour être détectables par le système de vision. Ce qui en fait des entités intrusives malgré le fait qu'ils augmentent la précision et la robustesse du système. Les seconds sont déjà présents dans le bâtiment dans des conditions normales. Il revient au système de vision de les détecter. Le choix de l'amer à détecter n'est pas anodin et a un impact sur les performances du système. Nous discutons de ce choix dans la section 7.2.2.

Les systèmes de SLAM, malgré leur performance, ne sont pas adaptés à notre problème car il ne rejoint pas notre objectif d'exploiter les capteurs déjà présents sur téléphone pour la plupart. Les techniques reposant sur les caméras RGB-D sont incompatibles avec les modèles de téléphones actuels due à l'absence de ses capteurs. L'erreur liée à l'estimation de la profondeur par caméra stéréo est inversement proportionnelle à la ligne de base (distance entre les 2 obturateurs). Cette distance est limitée par les dimensions du téléphone. Afin d'avoir une précision plus élevée pour l'estimation de la profondeur, la ligne de base doit être plus grande et donc les téléphones doivent être plus grands, ce qui à l'opposé de la tendance actuelle. Depuis

5 ans, la taille des téléphones ne dépasse pas les 7 pouces en diagonal [McNamara, 2021]. Le marché met même en vente des modèles plus petits. Ceci rend les solutions avec caméra 3D inattractives dans notre cas dû à leur incompatibilité avec l’objectif [F] du chapitre 1.

Pour les systèmes de SLAM visuel-inertiel utilisant une caméra 2D, ils restent sensibles aux environnements ayant des scènes dynamiques, aux problèmes de variations d’intensité et aux erreurs de détections et d’extractions de caractéristiques. Certains SLAM implémentent la détection de points clés et le traitement d’image sur des architectures hétérogènes CPU-FPGA. L’implantation de ces systèmes peut se révéler coûteuse et non compatible avec les modèles de téléphones actuels. Une autre problématique lors de l’usage d’autres capteurs est la positionnement de ce capteur sur le corps i.e. à quel endroit le porter pour qu’il soit stable et qu’il n’ait pas une partie de son champ de balayage occluse. Les LiDARs sont particulièrement sensibles à cette problématique. Enfin, les systèmes de SLAM peuvent monter une intensité calculatoire élevée et un impact mémoire très important. Ce qui peut rendre leur implémentation sur les téléphones actuels difficile voire impossible ne respectant pas l’objectif [F].

Les systèmes de VR utilisent des balises statiques pré-positionnées pour une solution clé en main où l’utilisateur porte un casque. Cela signifie que la zone où le système peut fournir une représentation fidèle de l’environnement est limitée par ces balises statiques. Bien que le coût de déploiement de ces balises soit raisonnable, l’incapacité à les mettre à l’échelle au niveau d’un bâtiment rend cette option insatisfaisante. Selon [de Souza Cardoso et al., 2020], la plupart des systèmes de AR se basent sur des marqueurs/amers. Il est nécessaire de placer des marqueurs artificiels ou d’identifier des marqueurs remarquables dans le bâtiment. Selon [Marchand et al., 2015], les solutions AR ont des limites de mise à l’échelle et d’acceptation par l’utilisateur. Ce sont clairement des domaines d’amélioration potentiels qui doivent être pris en compte à la fois par la recherche et l’industrie.

Une tendance claire apparaît dans toutes les approches citées précédemment. Elle concerne l’usage de modèles à apprentissage dans le but de détecter un objet que ce soit dans les systèmes SLAM, AR ou autres. Les modèles à réseau de neurones sont déployés pour effectuer cette détection car ces modèles ont montré des performances supérieures aux autres approches [Krizhevsky et al., 2012]. Même, ils connaissent leur propre limitation et amènent avec eux d’autres difficultés.

Quelque soit le modèle d’apprentissage à entraîner, il est absolument nécessaire de fournir une série d’exemples sur laquelle apprendre. Ces exemples étiquetés constituent une base de données d’entraînement. Ces bases de données doivent répondre à plusieurs critères en termes : de représentativité du problème traité, i.e. les exemples fournis décrivent précisément et fidèlement la tâche à effectuer ou la cible à identifier ; d’intégrité, i.e. les exemples fournis sont correctement étiquetés et formatés et exploitables par le modèle ; de disponibilité, i.e. le nombre et la variété des exemples sont suffisamment élevés pour éviter des biais lors de l’entraînement du modèle. Si ces bases de données n’existent pas, il est nécessaire de les créer.

Une famille de détecteurs d’objets basés sur des réseaux de neurones est YOLO. La position de YOLO dans l’état de l’art est présenté et le fonctionnement du modèle YOLOv5 est expliqué. Les différentes versions de YOLO ont été comparées dans l’état de l’art. YOLO montre de meilleures performances que ses concurrents. Il répond aussi aux contraintes sur la vitesse de calcul et l’espace mémoire liées au fait que notre système a pour objectif d’être déployable sur un téléphone. Dans cette deuxième partie de cette thèse, nous souhaitons exploiter les informations contenues dans la carte afin de produire un recalage visuel à partir des

amers détectés dans l'environnement intérieur. Et donc, le déploiement d'un réseau de neurones détecteur d'objet semble être la solution. Nous nous plaçons du point de vue utilisateur visant à exploiter un modèle neuronal pour corriger la position/cap du système proprioceptif par un signal extéroceptif. Le but est d'entraîner YOLO pour qu'il détecte un amer choisi dans le cadre de notre méthodologie pour être en accord avec l'objectif [D].

6.5 Structure de la partie II

Dans le chapitre 7, la présentation de *YOLO* en tant que détecteur d'objets basé sur réseau de neurones est faite. Une analyse haut niveau des blocs composants ce modèle neuronal est effectuée pour comprendre son fonctionnement et étudier l'adéquation de celui dans le cadre de la méthodologie proposée dans ce chapitre. Cette méthodologie concerne justement l'adaptation d'un modèle neuronal de détection d'objets pour un système de localisation. Son but est de décrire la réflexion autour du choix du détecteur et des amers à détecter, et de répondre aux questionnements qui découlent de ces choix concernant les données à utiliser pour l'entraînement du détecteur dans le cas où la quantité et la variété de données ne sont pas satisfaisantes pour permettre la convergence du modèle et effectuer un entraînement robuste.

Dans le chapitre 8, la mise en œuvre du recalage visuel dans notre système est présentée. Il s'agit d'instancier le détecteur afin de produire les détections qui sont le signal extéroceptif qui recalc visuellement le cap des particules du filtre particulaire. L'exploitation du modèle sténopé qui représente le fonctionnement de la caméra du téléphone permet d'obtenir l'angle dans lequel la détection réside ce qui fait de la détection une observation. Une phase d'appariement est nécessaire afin d'associer l'observation avec un amer déjà existant sur la carte du bâtiment. Chaque particule du filtre peut se recalculer sur l'amer qui lui est associé en incluant les contraintes spatiales tirées de la carte pour valider l'appariement. Une partie expérimentale sur le recalage visuel dans le cas où le détecteur ne produit que des vrais positifs est présentée. La dernière étape consiste à étudier l'impact de la détection d'une autre classe qui est fortement présente dans l'environnement intérieur et qui a présentée un cas limite auquel le système doit répondre. Un ré-entraînement du détecteur est nécessaire, ce qui implique de fournir les données adéquates pour cette classe là.

Dans un second temps, le détecteur d'objets peut produire des faux positifs. Ces fausses alarmes engagent le processus de recalage visuel dans des cas où l'appariement peut être erronée voire impossible. Donc, une campagne de tests est menée afin de qualifier les faux positifs qui peuvent être produits et tenter d'expliquer l'origine de ceux ci. Le but est de proposer des mécanismes de validation de détections en exploitant le lien entre l'amer à détecter, sa présence sur la carte du bâtiment et les hypothèses sur la détection. Dans ce cas, une gestion de la validité/non validité de la détection est implantée dans le filtre à particules où les particules sont groupées pour recevoir un traitement différent suivant si elles valident ou non la détection. Cette gestion des faux est testée par une expérience ayant des faux positifs.

6.5.1 Contributions

Nos contributions principales dans cette parties sont :

1. Proposition d'une méthodologie pour l'adaptation d'un détecteur d'objets pour le recalage visuel d'un système proprioceptif sur amer à identifier.

2. Exploitation d'un réseau de neurones convolutifs d'un point de vue utilisateur avec une approche boîte noire via un entraînement par transfert pour la mise en œuvre d'une correction de trajectoire par recalage visuel.
3. Validation des détections par l'application de contraintes géométriques tirées de la carte du bâtiment qui permet au PF une gestion de la possibilité d'un faux positif via des groupes dont le traitement est différencié [Ghaoui et al., 2023].

Chapitre 7

Détection et apprentissage pour la localisation

Sommaire

7.1	YOLOv5	132
7.2	Méthodologie de l'adaptation de YOLO pour recalage visuel . .	133
7.2.1	Méthodologie	133
7.2.2	Amers	134
7.2.3	Choix des classes d'amers	135
7.2.4	Base de données	136
7.3	Apprentissage par transfert	140
7.3.1	Congélation des couches	141
7.3.2	Entraînement du détecteur	142
7.3.3	Sur-apprentissage et arrêt prématuré	142
7.4	Évaluation	143
7.4.1	Métriques	143
7.4.2	Résultats	145
7.5	Faux Positifs et ré-entraînement de YOLO	147
7.5.1	Types de faux positifs	148
7.5.2	Ré-entraînement	150
7.5.3	Expériences	154
7.6	Récapitulatif	157
7.7	Conclusion	158

Dans le chapitre 6, il est cité que dans l'état de l'art, le recalage extéroceptif est souvent proposé pour corriger les trajectoires. Le recalage visuel se base sur un signal extéroceptif : l'image. Afin de réaliser un recalage visuel, il est nécessaire de définir quelle classe d'amers est considérée comme entité cible à détecter dans les images acquises et d'étudier le processus de détection. Il est aussi mentionné que les modèles de détections d'objets utilisant des réseaux de neurones montrent de meilleures performances. Il est nécessaire dans ce cas d'entraîner ce réseau afin qu'il puisse détecter la classe d'amer choisie.

Dans ce chapitre, la méthodologie d'adaptation du détecteur d'objet basé sur des réseaux de neurones est présentée. Cette méthodologie englobe, pour un détecteur déployé, le choix des classes d'amers à détecter, de la base de données pour entraîner ce détecteur et de l'approche d'apprentissage choisie. Ces points sont présentés et détaillés en finissant par l'instanciation et l'entraînement du modèle neuronal. Cette méthodologie est l'objectif [D] de ce manuscrit.

7.1 YOLOv5

Dans le chapitre 6, les algorithmes de la famille YOLO ont été présentés et comparés à leurs concurrents. YOLOv5 de [Jocher et al., 2020] est la cinquième itération du modèle de détection d'objets. Il représente une avancée majeure dans la capacité de l'intelligence artificielle à identifier et localiser les objets dans une image. Il a été conçu spécifiquement pour améliorer à la fois la vitesse et la précision de la détection d'objets par rapport à ses prédécesseurs, tout en minimisant les erreurs communes et les lacunes des versions antérieures.

Comme les versions précédentes de YOLO, YOLOv5 fonctionne en divisant d'abord une image en une grille. Chaque cellule de cette grille est chargée de prédire plusieurs boîtes englobantes et les scores de confiance associés à ces boîtes. Ces boîtes englobantes sont essentiellement des régions de l'image qui sont prédites pour contenir un objet particulier. Le score de confiance est une mesure de la certitude du modèle qu'une boîte englobante donnée contient réellement un objet et de la précision de la forme et de la taille de cette boîte.

Une fois ces boîtes englobantes prédites, YOLOv5 procède ensuite à la classification de ces régions en différentes catégories d'objets. Pour ce faire, chaque cellule de grille prédit également des probabilités de classe conditionnelles pour chaque boîte englobante. Ces probabilités indiquent la certitude du modèle qu'un objet donné appartient à une certaine classe.

Mais là où YOLOv5 se distingue vraiment, c'est dans les améliorations et les optimisations qu'il apporte à ce processus. Par exemple, YOLOv5 utilise plusieurs échelles de point d'ancrage pour les boîtes pour mieux gérer les objets de différentes tailles et formes. Cela permet au modèle de mieux gérer la diversité et la variabilité des objets qu'il peut rencontrer dans le monde réel.

Ces améliorations font de YOLOv5 un outil puissant pour la détection d'objets en temps réel. Sa vitesse et sa précision améliorées lui permettent d'être utilisé dans une gamme de scénarios d'intelligence artificielle encore plus large, allant de la surveillance vidéo en temps réel à la conduite autonome, en passant par la reconnaissance d'images pour l'Internet des Objets.

7.2 Méthodologie de l'adaptation de YOLO pour recalage visuel

7.2.1 Méthodologie

L'étude des détecteurs d'objet à partir d'image en utilisant des réseaux de neurones est un sous domaine de l'Intelligence Artificielle. L'entraînement et l'évaluation de ces modèles sont faits sur des jeux de données d'images collectées et pré-étiquetées. Les évaluations sont faites de manière à quantifier l'erreur que commet un modèle entre une détection et la vérité terrain. Ces notions sont purement virtuelles. Lors du déploiement d'un modèle dans un système de localisation en intérieur, le modèle fait face à une problématique concernant les phénomènes physiques ainsi que les limitations matérielles et les contraintes temporelles auxquelles est soumis le système de localisation.

La figure 7.1 illustre notre proposition pour une méthodologie sur l'adaptation du modèle YOLO pour recaler visuellement notre système proprioceptif présenté dans la première partie de ce manuscrit. Cette adaptation est la ligne directrice des choix du modèle à utiliser, des amers à détecter sélectionnés, des classes à entraîner, de la base de données à construire et de l'agencement de ce nouveau bloc de traitement dans le système de localisation.

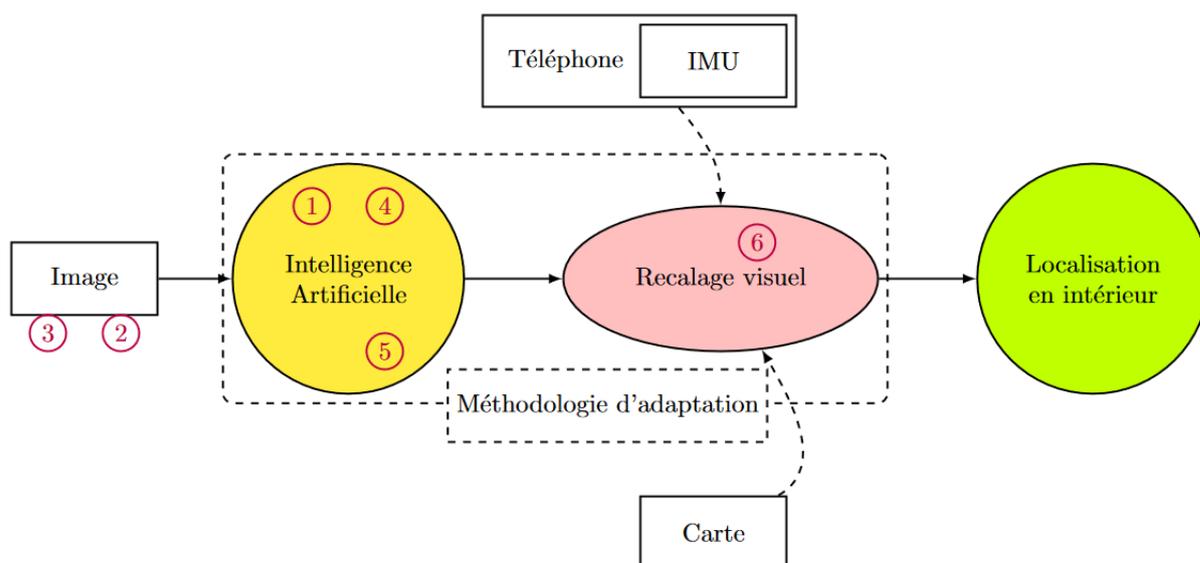


FIGURE 7.1 – Positionnement de notre proposition de méthodologie.

Le bloc "méthodologie d'adaptation" de la figure 7.1 se résume aux points suivants :

- ① Choisir parmi les modèles que propose l'état de l'art un modèle répondant à l'objectif [F] sur l'embarquabilité dans le téléphone (voir chapitre 6).
- ② Choisir les amers à détecter et identifier les classes de ceux-ci. Une analyse de la distinguabilité, densité et standardisation par rapport à la carte du bâtiment est nécessaire pour ce choix (voir ce chapitre).
- ③ Sélectionner ou créer le jeu données d'entraînement. Cette étape peut engager un surcoût en temps qui peut être problématique d'où l'importance d'un choix judicieux

des classes d'amers (voir ce chapitre).

- ④ Choisir la stratégie d'entraînement du détecteur d'objet. Ce choix est conditionné par le manque ou l'incomplétude du jeu de données. L'entraînement par transfert propose des résultats satisfaisants en entraînant le détecteur sur un jeu généraliste puis en le spécialisant sur le jeu de données dédié, ce qui réduit le coût d'entraînement (voir ce chapitre).
- ⑤ Mener une campagne de tests pour valider l'entraînement et trier les détections produites en isolant les cas problématiques (voir ce chapitre).
- ⑥ Décider pour chaque cas problématique s'il revient au détecteur de prendre en compte cette spécificité (par exemple, une classe non définie à détecter) ou si c'est au processus en aval de gérer ces cas problématique (voir chapitre 8).

D'un point de vu compromis vitesse et performance, les modèles YOLO ont montré de meilleures performances que leurs concurrents. Parmi les différentes versions de YOLO, YOLOv5 se présente comme le modèle adapté car ses performances et vitesse de précision dépassent les versions 1, 2, 3 et 4. *Ultralytics* propose des déclinaisons de YOLOv5 qui permettent d'effectuer la classification et la segmentation d'instance. Ceci sont soit plus lents soit présentent des cas problématique. Dans ce manuscrit, YOLO fait référence à YOLOv5 car l'intérêt est porté sur YOLOv5 dans le cas de la détection d'objet et de leur localisation dans une image, i.e. YOLO produit une boîte englobante autour de l'objet détecté dans l'image.

En termes de résolution depuis le tableau 6.1, la résolution 1280×1280 pixels est plus grande en largeur que la plupart des caméras de téléphones disponibles sur le marché dont la résolution est 1920×1080 pixels. De plus, une résolution plus importante induit un impact mémoire plus important, des temps de calcul significativement plus importants où, dans le tableau 6.1, YOLOv5x6 (1280×1280) est plus de deux fois plus lent sur GPU et environ quatre fois plus lent sur CPU que YOLOv5x (640×640). Donc, le modèle à la résolution 640×640 pixels est sélectionné. Et en terme de taille de réseau, YOLOv5x est sélectionné car il présente les meilleures performances dans un premier. En perspectives, une taille plus petite pourra être sélectionnée pour satisfaire notre objectif [F] concernant le déploiement de notre système dans un téléphone portable.

Dans le but d'effectuer le recalage visuel, YOLO doit détecter les amers présents dans l'espace intérieur. Pour cela, il est nécessaire de définir les amers en question et de procurer la base de données adéquate qui représente cet amer afin que YOLO puisse apprendre dessus. Notre méthodologie explicite le procéder de sélection de ces deux éléments ci-dessous.

7.2.2 Amers

Les amers sont des entités présentes dans l'espace physique 3D du bâtiment qui, une fois détectés, peuvent être transformés en observation. Cette observation est utilisée pour effectuer le recalage visuel. Le choix de l'amer à détecter est fonction de plusieurs critères :

— Présence :

Artificiel : dans le chapitre 6, il est mentionné que l'amer peut être artificiel i.e. un objet fabriqué pour le processus de localisation dans le but de le corriger. Cet amer artificiel est introduit dans l'espace en le plaçant à des endroits stratégiques. Dans ce cas, l'amer n'existe que pour cet objectif. Sa forme est étudiée dans le but de faciliter sa détection et de lui permettre d'encoder une information telle que sa position, son

identifiant, son type, ... Cet amer peut être considéré comme intrusif, coûteux, hors normes ou non esthétique par le propriétaire des lieux.

Organique : dans le cas opposé, l'amer organique est un objet existant déjà dans l'environnement en intérieur. Son existence et son placement ne dépendent aucunement du processus de localisation. Donc, contrairement à l'amer artificiel, celui-ci à ses propres raisons d'exister. Il permet de fournir le signal extéroceptif de manière opportuniste. La difficulté réside dans quels objets choisir.

- Identifiabilité : un amer doit être distinguable dans l'environnement, i.e. les confusions possibles entre un amer et un objet commun doivent être minimisées lors du choix de l'amer. Sinon, il reviendra au détecteur de développer des mécanismes de différenciation ce qui complexifient leur tâche. Ceux ci peut allonger les temps de traitements ou demander plus de ressources.
- Exploitabilité : en supposant que l'amer est détecté, il doit fournir une information spatiale et sémantique. L'information spatiale peut être composée de sa position dans l'image, de son orientation estimée à partir de l'image. L'information sémantique quant à elle peut être la classe de l'amer, son identifiant, ... Dans le cas où le recalage est conjugué avec une carte du bâtiment, comme dans notre cas, l'amer peut être positionné dans la carte du bâtiment en cherchant une correspondance avec les amers existants dans la carte. Cet appariement dans nos travaux est présenté au chapitre 8.
- Densité : l'amer choisi est présent en plusieurs exemplaires dans l'espace intérieur. Si il n'existe pas assez d'exemplaires, l'effet du recalage est négligeable car le système est, dans la grande majorité du temps, en mode proprioceptif. Dans le cas où il existe un nombre trop important d'exemplaires, la phase d'appariement est complexifiée, voit son temps d'exécution augmenter et peut même devenir impossible. Donc, il existe un équilibre à respecter sur la densité de cette classe d'amers dans l'espace intérieur.
- Standardisation : si l'amer est un objet standardisé, il combine systématiquement les points précédemment mentionnés. Ses aspects, formes, géométries et couleurs sont exactement les mêmes pour tous les exemplaires ce qui réduit la variance intra-classe de cet objet. Le fait que les attributs des amers soient identiques rend la détection de ceux-ci moins complexe, répétable d'un exemplaire à un autre. Mais, elle reste non triviale dans le cas d'un amer organique dû au fait que la projection de l'objet dans l'espace caméra dépend de son orientation dans l'espace 3D.

7.2.3 Choix des classes d'amers

Dans ce manuscrit, l'extraction d'information de la carte est un de nos objectifs qui permet d'améliorer le processus de localisation. Certains amers sont déjà mentionnés sur les cartes de plan d'étage du bâtiment. Ces mêmes amers sont disposés de manière à couvrir uniformément l'espace en intérieur à cause de leur usage [Mićović et al., 2021]. Ce sont les extincteurs.

Ils sont des objets standardisés et obligatoires dans chaque bâtiment. Ils ont une couleur et une forme détaillées par la norme européenne et ne doivent jamais être obstrués ni cachés. Ceci augmente leur détectabilité quand le piéton se déplace. Ces amers sont organiques car leur présence est obligatoire pour des raisons de sûreté et donc ils sont normés et non-intrusifs.

La mention de leurs positions sur le plan d'urgence est requise par les normes de sécurité. Et donc, il est possible de pré-positionner ceux-ci dans la carte du bâtiment pour les utiliser durant la localisation. Ceci en fait l'amer à détecter dans notre étude.

7.2.4 Base de données

Le modèle YOLO est entraîné sur le jeu de données Microsoft COCO [Lin et al., 2014] par ses auteurs [Jocher et al., 2020]. La version *train2017* de COCO possède 80 classes d’objet et est composée d’environ 123 287 images avec 886 284 instances. La liste des classes présentes et leurs décomptes dans COCO sont présentés en annexe C.3. Les auteurs de YOLO mettent à disposition ses poids pré-entraînés sur COCO sur 300 époques.

Dans la sous-section précédente, nous annonçons que l’extincteur est l’amer sur lequel le recalage visuel est effectué. Une première difficulté est que le jeu de données COCO ne possède pas la classe d’extincteurs. Donc, YOLO pré-entraîné ne peut pas détecter directement des extincteurs. Il est donc essentiel d’adapter la base de données pour qu’il puisse apprendre à reconnaître un extincteur. Cela signifie que la base de données devra contenir des exemples pertinents, représentatifs et intègres d’extincteurs. C’est la similarité entre ces exemples d’apprentissage et les images contenant des extincteurs durant la localisation qui a l’impact le plus important sur les performances du réseau de neurones.

C’est pour cela que la méthodologie d’adaptation impose de trouver ou construire un jeu de données avec les amers choisis. La construction de jeu de données est un processus coûteux en temps et en argent car il est nécessaire de collecter un nombre important d’images contenant les objets des classes choisies qui sont variées. Ensuite, il faut étiqueter chaque exemple dans chaque image par une boîte englobante et lui attribuer une classe. Cet étiquetage (*labelling*) fait à la main par des humains est l’étape la plus coûteuse dans la construction du jeu de données. Il existe parfois des cas où l’étiquetage n’est pas objectif, i.e. deux individus étiquetant la même image auront des boîtes différentes en géométrie et en sémantique. Dans notre cas, nous nous intéresserons ci-dessous au jeu de données *FireNet*.

7.2.4.1 FireNet

En 2019, FireNet a été publié par [Jadon et al., 2019]. C’est un jeu de données contenant des images d’équipement de sécurité. Il contient 1452 images étiquetées et segmentées (1452 × 3 fichiers) de 8 classes listées dans le tableau 7.1 ainsi que leur code Uniclass¹ qui est une spécification de l’industrie de la construction.

ID	Étiquette	Uniclass
1	Activateur d’alarme	Pr 40 50 28 64
2	Couverture anti-feu	Pr 75 75 30 50
3	Sortie	Pr 40 50 28 29
4	Extincteur	Pr 40 10 77 32
5	Signalisation d’extinction d’incendie	Pr 40 10 77 31
6	Orbes lumineux clignotants	Pr 75 75 30 97
7	Sirènes	Pr 75 75 30 30
8	Dômes blancs	Pr 75 75 30 65

TABLE 7.1 – Liste de classes présentent dans FireNet [Jadon et al., 2019].

Nous nous intéressons aux extincteurs, donc un script est mis en place pour éliminer toute étiquette et image qui ne contient pas d’extincteurs. Certaines questions liées à l’intégrité des étiquettes et au biais de l’expert se posent. Ces cas particuliers sont présentés ci-dessous et sont la cause de l’élimination de certains échantillons.

1. <https://uniclass.thenbs.com/>

7.2.4.2 Exemples rejetés

Double exemple Certains échantillons contiennent une étiquettes qui couvrent plusieurs extincteurs comme dans la figure 7.2b au lieu de l'étiquetage correct comme dans la figure 7.2a. La définition d'un extincteur n'est pas respectée dans ce cas. De plus pendant le recalage visuel, si l'on considère le centre de la détection comme le centre de l'amer, le centre de la boîte englobante en bleu n'est pas dans l'extincteur. Ceci peut conduire à des erreurs de recalage importantes. Ces exemples doubles sont éliminés du jeu de données.



(a) correct.



(b) incorrect.

FIGURE 7.2 – Étiquetage de deux extincteurs.

Extincteur caché La figure 7.3a contient un extincteur qui est caché dans un placard dédié avec un symbole indiquant sa présence. Dans notre cas, il ne s'agit pas d'un élément d'apprentissage valable en raison de l'absence de traits distinctifs décrivant un extincteur. Cette image montre une autre classe qui pourrait être "signalisation de la présence d'extincteur". Ainsi, nous filtrons tous les extincteurs cachés du jeu de données.



(a) Extincteur caché.



(b) Couleur autre que RAL 3000.

FIGURE 7.3 – Autres cas d'extincteurs.

Extincteur hors normes Les extincteurs dans l'UE et au Royaume-Uni suivent une norme qui définit leur forme, leur couleur et leur fonction². Cependant, le jeu de données FireNet semble contenir des extincteurs qui ne correspondent pas à la couleur RAL 3000 comme dans la figure 7.3b. Ces extincteurs ont été exclus de l'ensemble de données d'entraînement même si YOLO peut apprendre des caractéristiques indépendantes de la couleur des extincteurs.

7.2.4.3 Augmentation des données

Après avoir éliminé tout échantillon jugé non représentatif de la classe choisie, le nombre d'images est réduit à 196 images. L'augmentation des données est la technique qui permet d'artificiellement générer des échantillons supplémentaires à partir des échantillons initiaux. Cette opération ne remplace pas de véritables données acquises mais son efficacité a été prouvée pour le coût, purement logiciel, négligeable qu'elle propose [Mikolajczyk, 2021]. Elle est systématiquement déployée dans les entraînements. Elle permet aussi d'éliminer certains biais, i.e. si un jeu de données ne contient que des voitures bleues, le réseau de neurones peut apprendre que la couleur bleue est ce qui définit une voiture. L'augmentation peut modifier le jeu de données en ajoutant des exemples avec des canaux colorimétriques permutés. Toutes fois, l'augmentation peut introduire des biais ou fausser la donne en elle même telle qu'une symétrie horizontale sur un texte.

Dans notre cas, une première symétrie horizontale sur les images conservées est effectuée pour doubler le nombre d'images. Dans un seconde temps, le *framework Pytorch* augmente les données de formation par mosaïque. Cette augmentation dite en ligne est recommandée par les auteurs de YOLO [Jocher et al., 2020] car elle provient de leurs tests. Ils concluent que cette augmentation rend l'apprentissage plus robuste. Un exemple de cette mosaïque est donné par la figure 7.4.

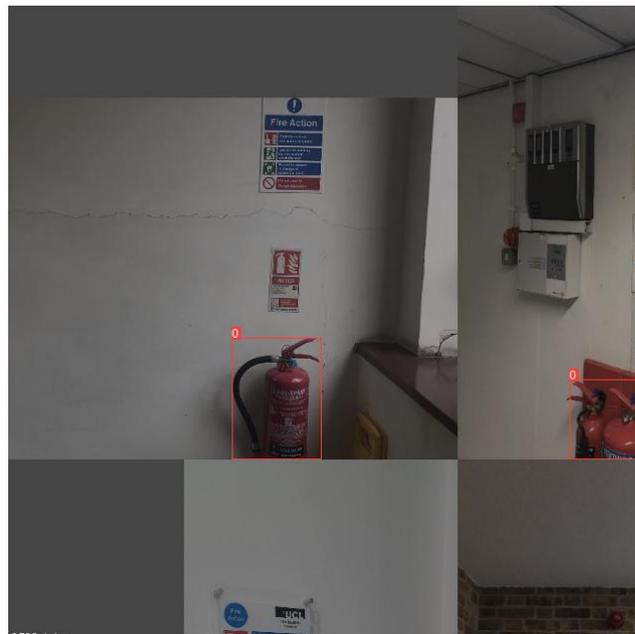


FIGURE 7.4 – Échantillon d'entraînement augmenté par mosaïque.

2. <https://www.standardsuk.com/products/BS-EN-3-7-2004-A1-2007>

7.2.4.4 Récapitulatif sur le jeu de données

Les spécifications du jeu de données ont été exposées. L'image à l'entrée du réseau de neurones est de résolution 640×640 . Le *framework* pré-traite les images pour obtenir la forme correcte. le nombre final d'images dans FireNet après épuration et augmentation est de 392 images. Même si le *framework* augmente les images en mosaïque durant l'entraînement, le nombre d'exemples est insuffisant pour entraîner un réseau de neurones. Comparé à COCO, il y a mille fois plus d'humains dans COCO que d'extincteurs dans le jeu de données résultants. Dans ce cas, il est recommandé d'effectuer un apprentissage dit par transfert.

Afin d'observer le jeu de données produit, les corrélogrammes des étiquettes avant augmentation sont montrés dans la figure 7.5. Cette image montre en abscisse la valeur relative de x, y, largeur et hauteur, et en ordonnées la hauteur, la largeur et y des boites englobantes. Les histogrammes ont une allure d'une distribution normale ce qui fait que les extincteurs dans les images sont normalement distribués. Il est possible grâce à cette figure de déduire s'il existe un biais dans les données i.e. tous les objets se trouve dans la partie gauche de l'image ce qui peut produire un biais sur l'entraînement qui impliquerait que "situer à gauche" soit une information pertinente. Ce qui ne l'est pas car il n'existe aucune raison que l'extincteur soit systématiquement situé dans la partie gauche d'une image quelconque.

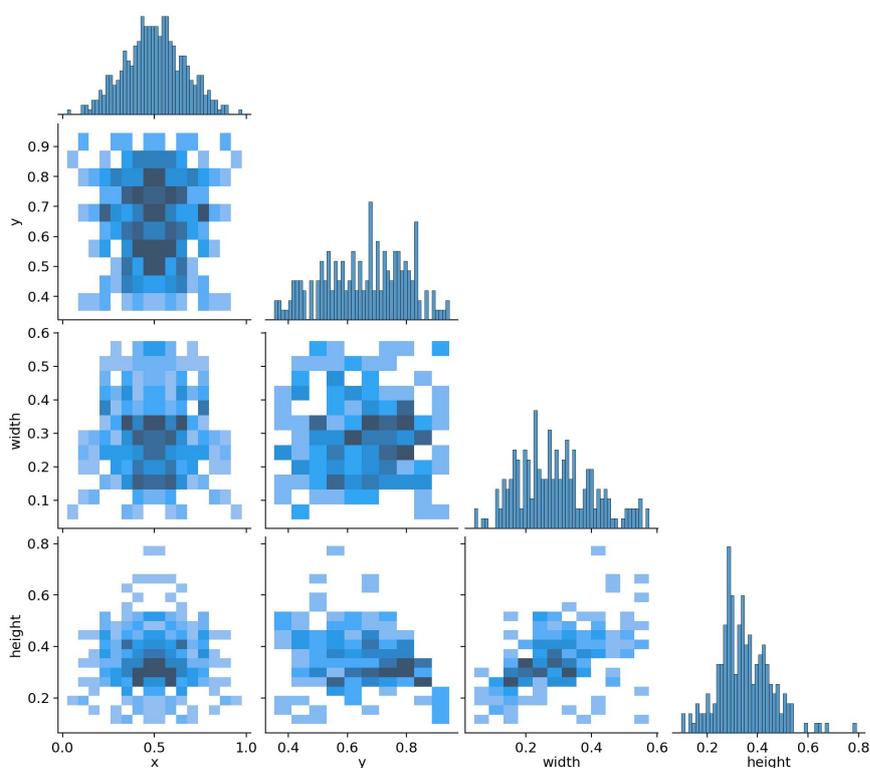


FIGURE 7.5 – Corrélogrammes des étiquettes.

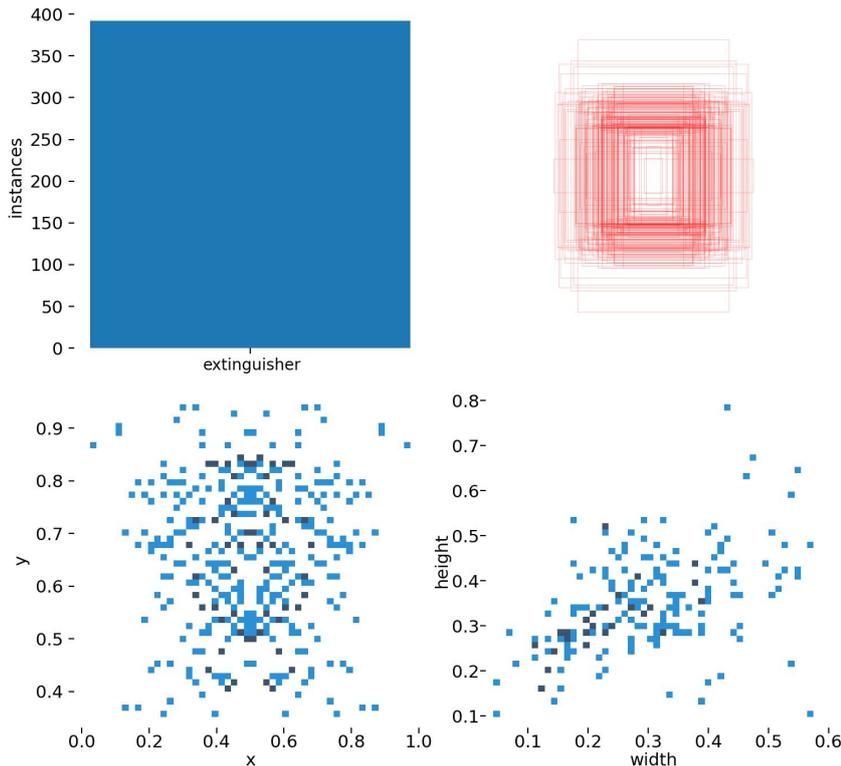


FIGURE 7.6 – Répartition des étiquettes.

La figure 7.6, Elle présente la répartition des instances sur les classes demandées. Nous n'avons que l'extincteur comme classe et donc il existe 392 instances de celui-ci. Les corrélations x vs y et largeur vs hauteur sont présentées. L'augmentation par symétrie horizontale fait que la corrélation x vs y est symétrique. Enfin, le tracé des boites englobantes superposées est fourni en rouge. Ce tracé permet de confirmer visuellement ce que les diagrammes de corrélation présentent. Ceci conclue la préparation du jeu données produit pour l'entraînement.

7.3 Apprentissage par transfert

L'apprentissage supervisé par transfert pour les réseaux de neurones profonds est une approche où un modèle préalablement entraîné sur une tâche est utilisé comme point de départ pour résoudre une autre tâche similaire. Cette méthode permet de transférer les connaissances acquises lors de l'apprentissage sur une tâche source vers une tâche cible, ce qui peut accélérer l'apprentissage sur la nouvelle tâche et améliorer les performances du modèle.

L'idée est que le modèle préalablement entraîné a déjà appris des représentations utiles des données, telles que des caractéristiques visuelles ou des connaissances générales sur la structure des données. En utilisant ces représentations initiales, le modèle peut se concentrer sur l'apprentissage des aspects spécifiques de la tâche cible, ce qui peut accélérer l'apprentissage et améliorer les performances.

L'apprentissage par transfert peut être réalisé de différentes manières, selon les similitudes et les différences entre la tâche source et la tâche cible. Différentes techniques existent telles que : le réglage fin (*fine-tuning*) où les poids du modèle préalablement entraîné sont ajustés en utilisant les données de la tâche cible ; ou la méthode d'extraction de caractéristiques où

les sorties d'une ou plusieurs couches intermédiaires du modèle source sont utilisées comme entrée pour un nouveau modèle cible.

L'apprentissage par transfert a montré son efficacité dans de nombreux domaines de l'apprentissage automatique, notamment la vision par ordinateur, le traitement du langage naturel et la reconnaissance vocale. Cette approche permet de réduire les besoins en données d'entraînement et de bénéficier des connaissances préalablement acquises pour résoudre de nouvelles tâches de manière plus efficace. Dans notre cas, le manque de données représentatives pour la classe extincteur nous pousse à entraîner YOLO par transfert selon les recommandations de ses auteurs [Jocher et al., 2020].

7.3.1 Congélation des couches

Les détecteurs d'objet à base de réseau de neurones suivent généralement une structure haut niveau découpable en deux blocs selon la figure 7.7. Un premier bloc est chargé de l'extraction de caractéristiques à partir de l'image fournie par une série de couches convolutives. C'est grâce à ces couches que ce bloc peut extraire les informations décrivant un objet. Un deuxième bloc chargé de la classification est composé de couches entièrement connectées. Ce bloc a pour rôle de déterminer à quelle classe l'objet présenté appartient et où il se trouve en se basant sur l'ensemble des caractéristiques extraites par le premier bloc. Ce deuxième bloc apprend les combinaisons de caractéristiques permettant de décrire l'objet en question. En apposant, la figure 7.7 à la figure 6.2, on peut identifier que "Head" + "Detection" de YOLO sont le second bloc.

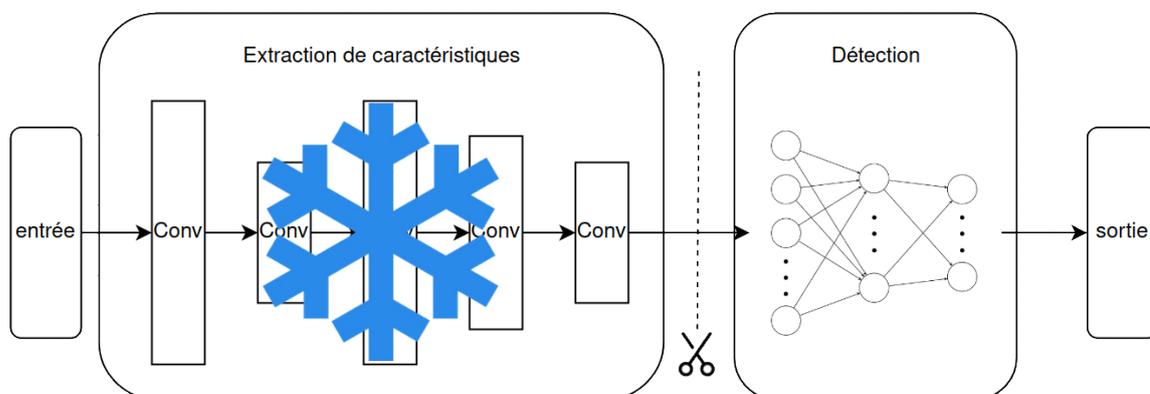


FIGURE 7.7 – Procédure d'entraînement par transfert : congélation des couches convolutives + ré-entraînement des couches de détection ("Head+Detect").

Afin d'effectuer l'apprentissage par transfert, il faut congeler les poids des couches convolutives. Et en même temps, il faut sectionner la partie détection et remplacer les couches de classification existantes par celles des classes désirées et ensuite procéder à l'entraînement des nouvelles couches de classification sur notre jeu de données.

Le but de la congélation est de préserver la connaissance déjà contenue dans les neurones convolutifs car ceux-là "savent" déjà décrire un objet. Il suffit d'indiquer au modèle quelles connaissances sont utiles pour décrire la classe cible d'où la nécessité de ré-entraîner les couches de classifications.

7.3.2 Entraînement du détecteur

Le diagramme en figure 7.8 présente le processus de rétro-propagation du gradient ("*gradient backward propagation*"). Durant l'entraînement, pour une entrée, le réseau infère une prédiction sous la forme d'une boîte pour chaque objet, de score d'appartenance à chaque classe et d'un score sur l'objectivité de la détection. La perte (*loss*) est calculée comme une fonction de cette prédiction contre la vérité terrain étiquetée par l'humain. Le gradient est rétro-propagé vers les couches précédentes où chaque neurone va ré-évaluer son propre poids en fonction de ce gradient afin de converger et de produire de meilleures prédictions. Les détails du calcul de ces deux grandeurs sont en dehors de la portée de ce manuscrit mais il est nécessaire de connaître leur arrangement dans le diagramme afin de comprendre l'impact de ceux-ci et de valider le fait que le modèle apprend.

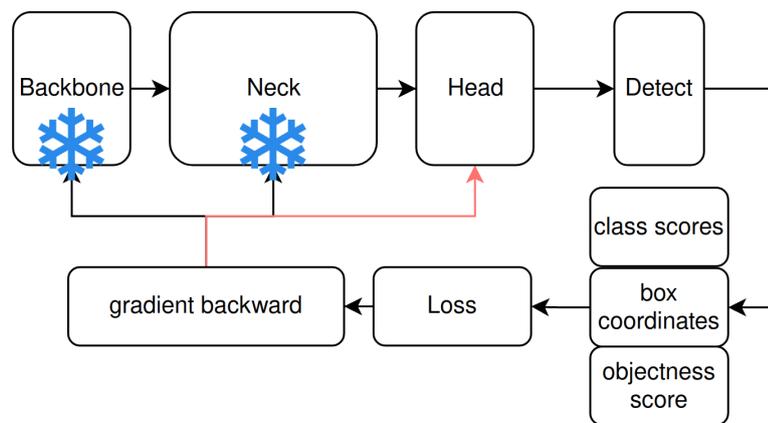


FIGURE 7.8 – Entraînement de YOLO basé sur le diagramme de la figure 6.2.

Le fait d'avoir congelé les sous blocs "*Backbone*" et "*Neck*" fait qu'aucune modification de leur poids n'est possible. Par contre "*Head*" est bien modifié. Donc image par image, le réseau affine les poids cette couche. À la fin d'une époque d'entraînement, les nouveaux poids calculés sont mis en place pour refaire ce cycle dans l'époque suivante. Le but est de faire baisser la fonction de perte à chaque époque jusqu'à atteindre une valeur qui est jugée suffisamment basse.

7.3.3 Sur-apprentissage et arrêt prématuré

L'un des phénomènes le plus compromettants lors de l'entraînement est le sur-apprentissage. C'est le phénomène où le modèle finit par apprendre des éléments qui ne sont pas considérés comme une information décrivant l'objet en question. Le modèle finit par apprendre le bruit contenu dans la donnée. Ce qui rend le modèle très performant sur son jeu de données. Mais rend aussi la généralisation à des données autres que celles de l'entraînement infaisable et la performance de détection durant les tests baisse significativement. La prévention de ce phénomène est faite soit en augmentant la quantité et la variété des données, ce qui n'est pas faisable dans notre cas, ou bien en surveillant les performances du modèle via un jeu de données de validation.

En calculant les performances après X époques, la métrique de performances sur le jeu d'entraînement ne fait qu'augmenter, alors que sur le jeu de validation, cette même métrique

aura nécessairement une allure parabolique. L'arrêt prématuré vise à stopper l'entraînement quand la métrique de validation cesse de diminuer et à sauvegarder les poids aux meilleures performances sur les X dernières époques.

7.4 Évaluation

Les réseaux de neurones sont évalués en utilisant une partie du jeu de données que le réseau n'a pas eu en entraînement ni en validation. Le jeu de données test permet de calculer des métriques de performances dans un cas très proche du cas réel. Il représente une très petite partie ($\approx 5\%$) du jeu de données total. Pour quantifier les performances du réseau de neurones, il est nécessaire de calculer plusieurs métriques. Ces métriques sont conditionnées par des critères d'acceptabilité concernant la comparaison entre la prédiction du réseau de neurones contre la vérité terrain. C'est pour cela que leurs définitions sont présentées ci-dessous.

7.4.1 Métriques

7.4.1.1 Précision

La précision (P) mesure la proportion d'instances prédites comme positives qui sont réellement positives. Elle est calculée en divisant le nombre de vrais positifs (VP) par la somme des vrais positifs et des faux positifs (FP). La formule de la précision est donnée par l'équation (7.1) :

$$P = \frac{VP}{VP + FP} \quad (7.1)$$

La précision évalue la qualité des prédictions positives du modèle, en indiquant combien de fois le modèle est correct lorsqu'il prédit une classe positive. Une précision élevée indique que le modèle fait peu de fausses alarmes.

7.4.1.2 Rappel

Le rappel (R), également appelé sensibilité ou taux de vrais positifs, mesure la proportion d'instances positives réellement détectées par le modèle. Il est calculé en divisant le nombre de VP par la somme des vrais positifs et des faux négatifs (FN). La formule du rappel est donnée par l'équation (7.2) :

$$R = \frac{VP}{VP + FN} \quad (7.2)$$

Le rappel évalue la capacité du modèle à identifier toutes les instances positives réelles. Un rappel élevé indique que le modèle a une capacité élevée à trouver les instances positives, minimisant ainsi les cas de manque de détection.

7.4.1.3 F1

Le score F1 est une mesure combinée de la précision et du rappel, qui permet d'évaluer globalement les performances d'un modèle. Il est calculé en prenant la moyenne pondérée

harmonique de la précision et du rappel, donnant ainsi plus de poids aux valeurs plus faibles. La formule du F1 est donnée par l'équation (7.3) :

$$F1 = \frac{2 \times (P \times R)}{P + R} \quad (7.3)$$

Le F1-score est particulièrement utile lorsque l'on souhaite trouver un équilibre entre la précision et le rappel. Il est souvent utilisé lorsque les classes positives et négatives sont déséquilibrées dans les données.

7.4.1.4 IoU

L'*Intersection over Union* (IoU), également appelé *Jaccard Index*, est une mesure de similarité entre deux ensembles, généralement utilisée pour évaluer la qualité de la segmentation ou de la détection d'objets pour un traitement d'image. Elle est calculée en mesurant le rapport de l'intersection entre la prédiction et le vrai objet par rapport à leur union. L'IoU est calculée en divisant l'aire de l'intersection par l'aire de l'union des deux régions. Plus l'IoU est proche de 1, plus la prédiction est considérée comme précise et similaire au vrai objet. L'IoU est donc une mesure de similarité et le seuil à dépasser fixé pour celle-ci décide si la prédiction est un vrai positif ou faux positif.

$$IoU = \frac{S_{det} \cap S_{vrai}}{S_{det} \cup S_{vrai}} \quad (7.4)$$

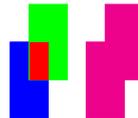


FIGURE 7.9 – Visualisation de l'intersection (rouge) sur l'union (magenta).

7.4.1.5 mAP

Le score *mean Average Precision* (mAP) est une mesure couramment utilisée pour évaluer les performances des modèles de détection d'objets. Elle est calculée en calculant la précision moyenne pour différentes valeurs de seuils de similarité (généralement basées sur l'IoU) et en prenant la moyenne de ces précisions. La précision est calculée en comparant les prédictions du modèle avec les annotations de vérité terrain, et en considérant une détection correcte lorsque l'IoU dépasse un seuil prédéfini. Le score mAP mesure à la fois la précision et le rappel du modèle, en tenant compte des fausses alarmes et des objets manqués. Le score mAP évalue ainsi la qualité de la détection d'objets.

Pour K classes, la mAP est définie par l'équation (7.5).

$$mAP = \frac{1}{K} \sum_{i=0}^K AP_i \quad (7.5)$$

Dans ce cas AP est *Average Precision* et est calculée en calculant d'abord la précision (P) à différents seuils de rappel (R), puis en prenant la moyenne de ces précisions. Le seuil de rappel correspond au taux de vrais positifs à partir duquel une prédiction est considérée comme positive. Pour N seuils de R, l'AP est donnée par l'équation (7.6).

$$AP = \sum_{i=1}^{i=N} (R_{i-1} - R_i) P_i \quad (7.6)$$

Dans le cas mono-classe, la mAP devient simplement l'AP. Dans les évaluations, deux mAPs sont mentionnées : mAP50-95 et mAP50 qui sont mesurées pour différentes IoU.

mAP50 se réfère à la "précision moyenne à un seuil IoU de 0,5". Cette métrique calcule l'AP en utilisant uniquement un seuil IoU de 0,5. mAP50 évalue la précision des modèles de détection d'objets en considérant principalement la détection d'objets qui ont un recouvrement significatif avec les vérités terrain. Cette métrique est couramment utilisée pour évaluer les performances des modèles de détection d'objets, en particulier dans des scénarios où un seuil IoU de 0,5 est considéré comme un critère de réussite adéquat pour la détection.

mAP50-95, d'autre part, signifie "moyenne de la précision moyenne entre les seuils IoU de 0,5 et 0,95". Cette métrique calcule l'AP en utilisant une plage de seuils IoU allant de 0,5 à 0,95, avec un pas de 0,05. Elle évalue les performances du modèle sur un large éventail de seuils d'IoU, ce qui permet de prendre en compte différentes tolérances en termes de recouvrement entre les prédictions et les vérités terrain. mAP50-95 est généralement utilisée pour évaluer la précision de localisation dans l'image des modèles de détection d'objets, en prenant en compte une variation de la précision sur différentes régions d'intersection entre la prédiction et la vérité terrain.

7.4.2 Résultats

La procédure d'entraînement de YOLO est effectuée sur 300 époques afin de voir l'évolution des différentes métriques présentées ci-dessous. Grâce à l'arrêt prématuré, il est possible de sélectionner une version considérée convenable parmi les 300 époques.

Les figures 7.10a et 7.10b présentent les évolutions des précision et rappel en fonction de la confiance dans la détection, respectivement. La courbe précision montre qu'il est possible d'inclure tous les VP vus pendant l'entraînement si le seuil de confiance est placé à 0,89. La courbe R par contre n'a pas de valeur de seuil de confiance pour laquelle aucun FP n'est produit.

Dans les figure 7.11a, la capacité du détecteur à distinguer les vrais positifs des vrais négatifs pour un seuil IoU de 0.5. La courbe s'approche du point (1,1). Cela implique que notre détecteur a une très grande capacité à différencier les VP des VN.

Dans la figure 7.11b, la courbe représente "l'intersection" des deux courbes en figures 7.10a et 7.10b où une confiance trop basse implique que le détecteur ne peut distinguer les VP. Pour une confiance trop élevée, le nombre de FN devient trop important. Pour le jeu de données sélectionné, une confiance de 0.316 en la détection est le seuil minimum pour avoir un compromis correct..

La figure 7.12 présentent de gauche à droite :

- *box loss* : moyenne par époque liée à l'entraînement et à la validation. Elle présente à quelle point la boîte englobante prédite englobe correctement l'objet comparée à la vérité terrain.
- *obj loss* : moyenne par époque qui exprime à quel point l'objet détecté est cohérent en combinant la classe prédite et la boîte englobante prédites.
- *cls loss* : cette métrique indique l'évolution de la perte interclasse. Dans notre cas, le détecteur ne connaît l'existence que d'une seule classe : l'extincteur. Donc cette

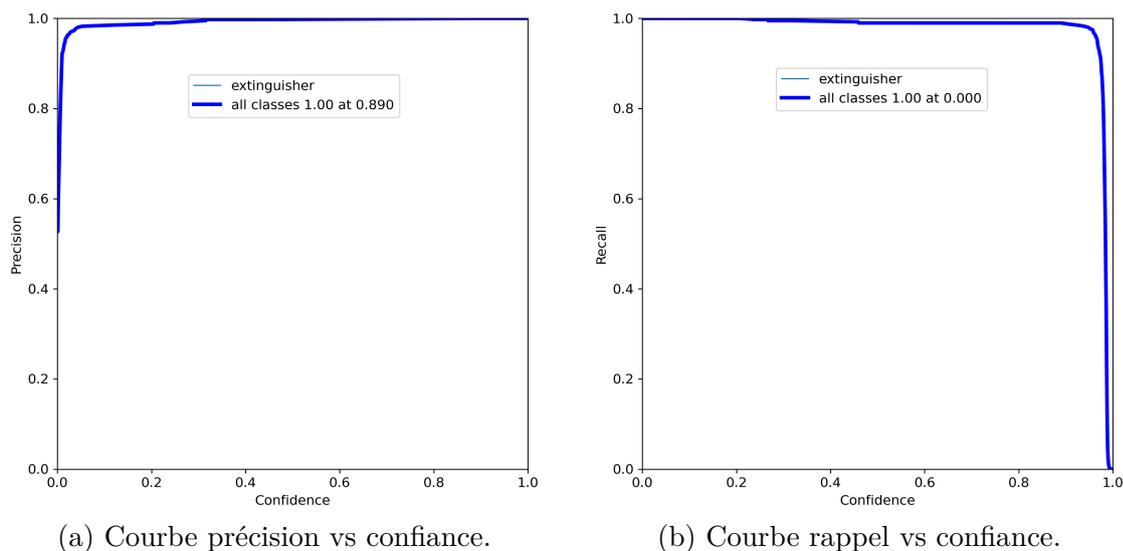


FIGURE 7.10 – Courbes de performances (1)

métrique est incalculable.

- *Précision* : cette métrique augmente au fur des époques montrant que le détecteur apprend à mieux distinguer les VP des FP.
- *Rappel*, de même que pour la précision, cette métrique augmente au fur des époques car le détecteur apprend à mieux détecter des VP des FN.
- La $mAP@0.5$ et $mAP@0.5 : 0.95$ sont présentées. La première évolue beaucoup plus vite que la deuxième car elle est relativement plus simple à satisfaire alors que la deuxième est plus exigeante.

Les différents éléments présentés après cet entraînement montrent que notre détecteur a appris à détecter les extincteurs. Le positionnement de la boîte englobante dans l'image affecte l'IoU. Donc les performances de ce détecteur se définissent en fonction de ce qui est jugé acceptable en terme d'IoU. L'IoU a un impact important sur le recalage visuel car, si la boîte englobante est décalée, l'angle dans lequel l'amer est vu change.

Dans ce cas, la $mAP@0.5 : 0.95$ est la métrique clé. Dans la figure 7.12, sa courbe se rapproche autour de l'époque N#200 puis semble ne pas dépasser significativement l'asymptote en y pour cette valeur là. Cela permet de sélectionner les poids à la deux centième époque pour le déploiement dans le système de localisation.

D'autres parts, le fait qu'il n'existe pas de valeur seuil pour que le rappel soit à un implique que notre détecteur aura plus de difficulté à gérer les FN. On peut s'attendre à manquer des amers lors du mouvement.

Enfin, les performances montrées dans ce chapitre sont liées au jeu de données construit pour entraîner, valider et tester YOLO pour la détection et localisation dans l'image d'un extincteur qui servira d'amer. Durant le déploiement dans le système de localisation, les performances du détecteur seront inférieures ou égales à celles présentées ici. Car, les performances d'un modèle basé sur des réseaux de neurones dépendent très fortement de la corrélation entre le jeu de données d'entraînement et les données acquises durant le déploiement. Les différences qui peuvent exister entre ceux ci peuvent être matérielles, i.e. le cas où les caméras ont des paramètres optiques différents. FireNet a été collecté par ses auteurs avec un téléphone. Il rejoint donc notre objectif de n'exploiter que des données déjà disponibles sur téléphone.

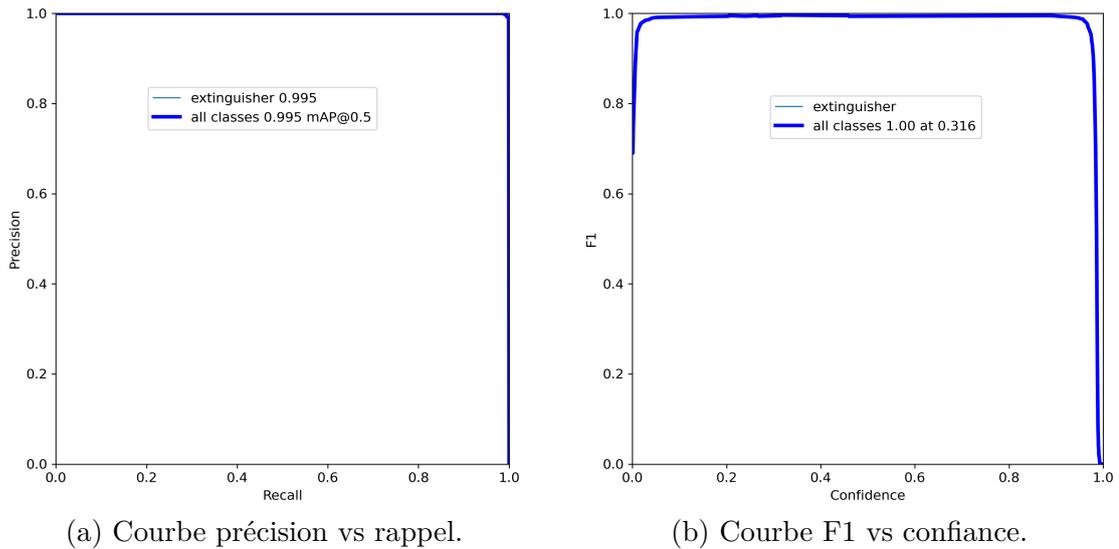


FIGURE 7.11 – Courbes de performances (2)

D'autres facteurs, comme la luminosité et l'exposition, peuvent aussi être la cause de réduction de performances même si les réseaux de neurones sont moins sensibles à ces phénomènes.

Il est aussi important de noter que malgré le fait que les extincteurs soient standardisés, il est possible d'avoir des différences sur les modèles ou les versions présentes entre les extincteurs du jeu de données et ceux de l'environnement intérieur lors du déploiement. Ceci implique que la variance intra-classe de la classe extincteur est certes faible mais non nulle. Si ce raisonnement est étendu, il existe au moins deux types d'extincteurs qui diffèrent assez par leur forme : l'extincteur à poudre et l'extincteur à eau/CO₂. Dans FireNet, ces extincteurs sont confondus en une seule classe. Il est possible de changer cette classification en deux classes distinctes, mais cela implique de ré-étiqueter tous les exemples à la main ce qui est une opération coûteuse. Mais, aussi, cela demande encore plus d'exemples car il faut apprendre à YOLO la différence entre les deux extincteurs déjà très similaires. C'est pour ces raisons que l'on a considéré qu'il n'existait qu'une seule classe d'extincteurs.

7.5 Faux Positifs et ré-entraînement de YOLO

L'entraînement du détecteur d'objets YOLO vise à minimiser l'erreur entre les prédictions de YOLO et les exemples fournis pendant l'entraînement. Les performances de YOLO sont liées à la similarité entre ces exemples et les images collectées lors du déplacement. Le détecteur n'étant pas parfait, des erreurs de détections se produisent sous la forme de faux positifs qui produisent des fausses alarmes. Si ces fausses alarmes ne sont pas gérées, le recalage visuel sera invoqué sur une donnée fausse et l'appariement d'une particule avec un amer peut n'avoir aucune solution pour aucune particule, i.e. toutes les particules doivent être détruites. Il est aussi possible que l'appariement soit sur un amer incorrect ce qui va fausser les particules.

Afin d'identifier les images posant problèmes, une campagne d'acquisition vidéo composée dans notre bâtiment est menée. Cette campagne est composée de 4893 images qui sont ensuite fournis à l'entrée de YOLO afin de produire des détections. Un faux positif est défini par le fait que le détecteur d'objets produit une détection dans une zone où l'élément cible n'est pas

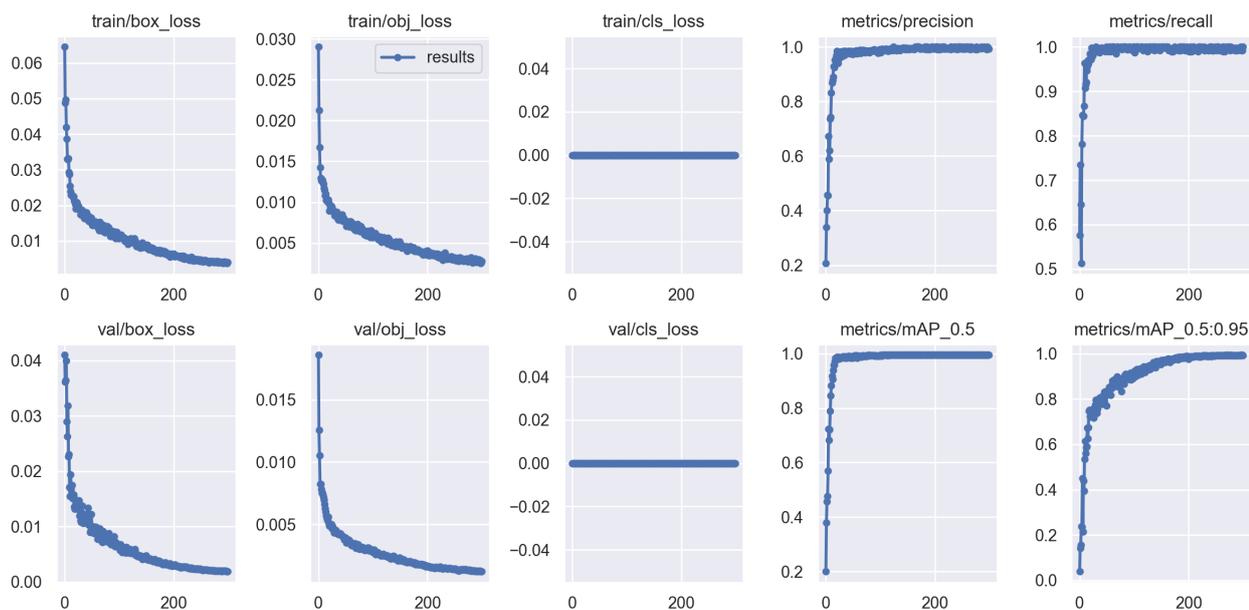


FIGURE 7.12 – Courbes d'évolution des métrique de l'entraînement sur 300 époques.

entièrement situé ou une partie n'ayant pas été détectée ne représente pas la vérité terrain.

7.5.1 Types de faux positifs

Bouteilles La figure 7.13 montre des faux positifs qui sont des bouteilles avec parfois une confiance relativement élevée. Les classes bouteille et extincteur ont plusieurs similarités entre elles en termes de forme et morphologie. Quand la couleur est similaire, ces similarités s'accroissent. Mais surtout, la détection multi-échelle de YOLO est plus sensible quand l'objet est considéré petit, i.e. occupe une surface en pixel faible. ce qui est équivalent à soit un objet 3D de petites dimensions ou un objet éloigné en profondeur de l'obturateur.

Reflets La présence de vitres et surfaces réfléchives dans les bâtiments fait que YOLO peut détecter un reflet comme étant un objet cible. Ce genre de faux positifs n'est pas trivial à résoudre du fait que les reflets copient les caractéristiques objets originaux au point où, aux sens des descripteurs, il peuvent être considérés comme de véritables objets de la classe cible. Il n'existe pas de moyen propre d'indiquer au réseau de neurones que la surface est réfléchive sans tirer partie des informations géographiques fournies par la carte. Donc au sens de la localisation, le reflet est strictement un faux négatif.

Superposition Ce phénomène appelé ainsi car il présente le cas où plusieurs extincteurs sont pris dans la même boîte de détection comme montré dans la figure 7.14b. Ce résultat se produit car l'ensemble de descripteurs nécessaires est partagé entre les deux extincteurs. Donc, le modèle déduit qu'il s'agit d'un extincteur plus grand.

Détections aberrantes L'usage de l'apprentissage par transfert au chapitre 7, implique qu'il peut y avoir soit des caractéristiques d'éléments appris durant la première phase d'entraînement sur le jeu de données généraliste ; soit des similarités non évidentes entre les classes



FIGURE 7.13 – Plusieurs exemples de faux positifs.

antécédentes et actuelles. Dans la figure 7.15a, le détecteur prédit à 82% de confiance qu'il s'agit d'un extincteur. Il est difficile de comprendre pourquoi cette détection est produite avec tant de confiance. Plusieurs hypothèses sont possibles la plus probable est liée aux 3 tailles de détection des cibles de YOLO : petit, moyen, grand. Ici, la cible est considérée petite voire moyenne, ce qui représente des cas plutôt difficiles pour YOLO. La détection de petite cible elle-même est une problématique non résolue en apprentissage [Liu et al., 2021].

Détections inexplicables Les cas de FP comme ceux de la figure 7.15b sont difficiles à expliquer et le manque d'explicabilité est l'un des inconvénients de l'utilisation du réseau neuronal. Pour cette raison, l'explicabilité est un domaine de recherche très actif au sein de la communauté de l'intelligence artificielle [Yuan et al., 2022, Joshi et al., 2021, Burkart and Huber, 2021]. Notre objectif n'est pas de comprendre pourquoi le FP s'est produit ni d'étudier l'explicabilité du réseau de neurones. Il s'agit plutôt de prendre en compte la possibilité d'un tel faux positif inexplicable et de s'assurer que notre système est robuste face à lui. Cela signifie que nous cherchons à quantifier la confiance de notre système en ces détections au moment où une observation se produit. De manière plus générale, ces détections inexplicables sont difficiles à quantifier et représentent le plus grand danger si le système de localisation se reposait uniquement sur ces détections.

Surfaces transparentes Il existe une catégorie de détections qui peut se produire malgré la séparation physique entre l'amer et la caméra. Ce sont les cas où l'amer se trouve derrière un mur "transparent" ou une forme de grillage. La figure 7.16 montre un exemple de ces détections qui sont valides mais impossibles à traverser. Ces FPs sont triviales à résoudre en ajoutant une couche sémantique à la carte du bâtiment indiquant que un mur est transparent ou opaque.



(a) Exemple d'une détection d'un reflet.



(b) Détection superposée pour 2 extincteurs dans une même boîte.

FIGURE 7.14 – D'autres exemples de faux positifs.



(a) Faux positif où un humain est détecté comme un extincteur.



(b) Détection inexplicable.

FIGURE 7.15 – D'autres exemples de faux positifs.

Faux négatifs Parmi toutes les images fournies en entrée, le modèle produit une quantité relativement importante de faux négatifs. YOLO possède 3 niveaux de détections : petit, moyen et grand. Cela lui permet de prédire la présence d'un objet en fusionnant l'information venant de ces 3 niveaux. À quelle distance peut-on affirmer que l'extincteur est détectable? La figure 7.17 donne une exemple du même extincteur à différente profondeur dans le champ. La question est : parmi les 4 images, dans les quelles voit-on un extincteur? Les faux négatifs dans notre cas ne sont pas traités et leur traitement est laissé en perspective. Cela implique que quand le système manque une alarme, la navigation reste proprioceptive.

L'analyse des faux positifs montre la présence d'humains parmi les détections. La spécialisation lors de l'entraînement au niveau du processus de détection d'objets a été trop importante. Il est nécessaire de revenir à la méthodologie d'adaptation afin de voir s'il existe une possibilité de permettre au réseau de ne pas confondre les classes humains et extincteurs.

7.5.2 Ré-entraînement

Afin de permettre à YOLO de détecter la classe d'objet humain dans les images, il est nécessaire que YOLO apprennent l'ensemble des descripteurs. L'entraînement par transfert



FIGURE 7.16 – Exemple de détections à travers des murs "transparents".



FIGURE 7.17 – Images contenant le même extincteur prises à différentes profondeurs de champ.

requiert de détruire les couches de classification pour ensuite les remplacer par un classificateur mono-classe. Ce qui fait que l'information concernant la classe humain est perdue. Une première possibilité est d'intervenir avant de détruire les couches de classification pré-entraînée sur COCO en sauvegardant les poids de la classe humain. Ensuite, ces poids sont réinstallés dans un modèle bi-classes. Cette technique a l'avantage d'économiser le temps et le coût de l'entraînement du réseau de neurones. Ses désavantages se présentent en deux : cette démarche demande une intervention au très bas niveau de l'implémentation de YOLOv5. Il est intuitif de penser qu'il est possible d'entraîner les deux ensembles de poids séparément. Mais de plus, il est difficile de prédire les effets d'un entraînement où seuls des extincteurs sont présentés pour un classificateur bi-classe à qui les poids de la deuxième classe sont joints, surtout quand les poids de cette classe ont été entraînés avec certains paramètres différents comme le nombre d'époques.

L'autre méthode est de ré-entraîner YOLO par transfert mais cette fois en formant un jeu de données possédant les exemples de chacune des deux classes. On combine alors le jeu de données *FireNet* qui est dédié aux extincteurs que l'on a épuré avec une version épurée de COCO [Lin et al., 2014] en sélectionnant uniquement les exemples humains qui sont existants dedans. Précédemment dans ce chapitre, il est discuté que le choix de l'extincteur comme

amer à détecter visait à minimiser la variance intra-classe. L'humain est l'une des classes ayant la plus grande variance intra-classe. Dans l'annexe C.1, l'humain est de loin la classe la plus représentée dans COCO (six fois plus que la deuxième classe). Ceci implique que l'humain est une classe difficile à détecter sur laquelle peu d'*a priori* tiré de la localisation peut être exploité et sur laquelle la carte du bâtiment ne fournit aucune information. Cette classe est donc problématique, fausse certaines détections et demande plus d'attention lors de l'entraînement. Il est donc naturel qu'elle exige plus d'exemples pour que le détecteur d'objet puisse apprendre à reconnaître celle-ci.

Le jeu de données contenant les deux classes est construit. La figure 7.18 montre que dans ce jeu, 392 instances d'extincteurs sont présentes contre plus de 1600 humains. Aussi, la répartition de ces humains est uniforme en colonnes de l'image et gaussienne centrée autour des lignes milieux de l'image.

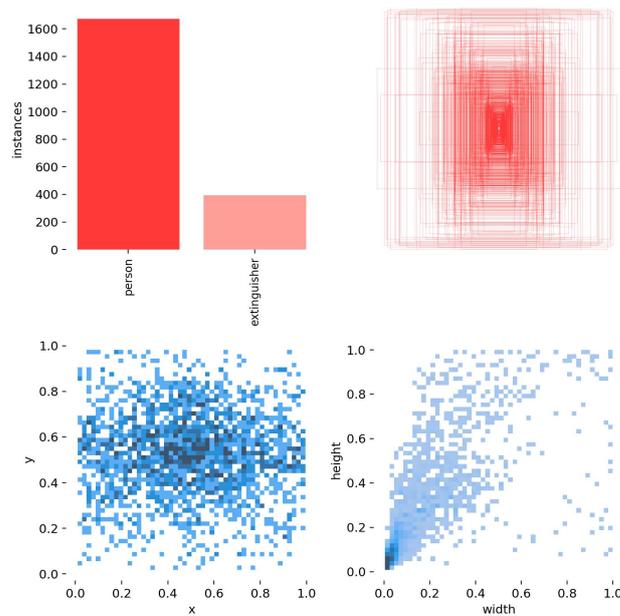


FIGURE 7.18 – Répartition des étiquettes pour les deux classes : humain et extincteur.

L'entraînement de YOLO sur ce jeu de données produit les courbes de la figure 7.19. Comparées à celles de la figure 7.12, les courbes pour la perte interclasse (*cls_loss*) sont non nulles du fait qu'il existe plus d'une classe. La courbe de *mAP 0.5 : 0.95* monte plus lentement dans ce cas. En examinant la figure 7.20, les courbes de précision et rappel sont présentées pour les 2 classes et la précision et rappel moyens. Ces figures montrent que le détecteur d'objet a plus de facilité à détecter correctement les extincteurs que les humains dû aux fait que ces derniers sont plus variés en formes et couleurs.

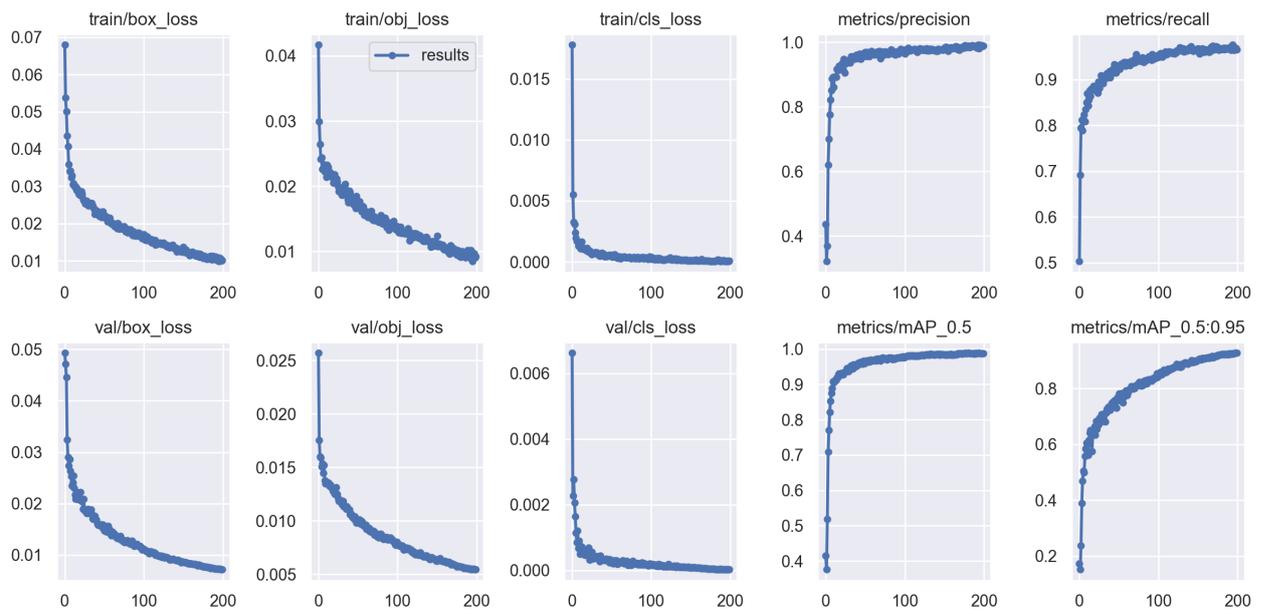


FIGURE 7.19 – Courbes d'évolution des métrique de l'entraînement sur 200 époques.

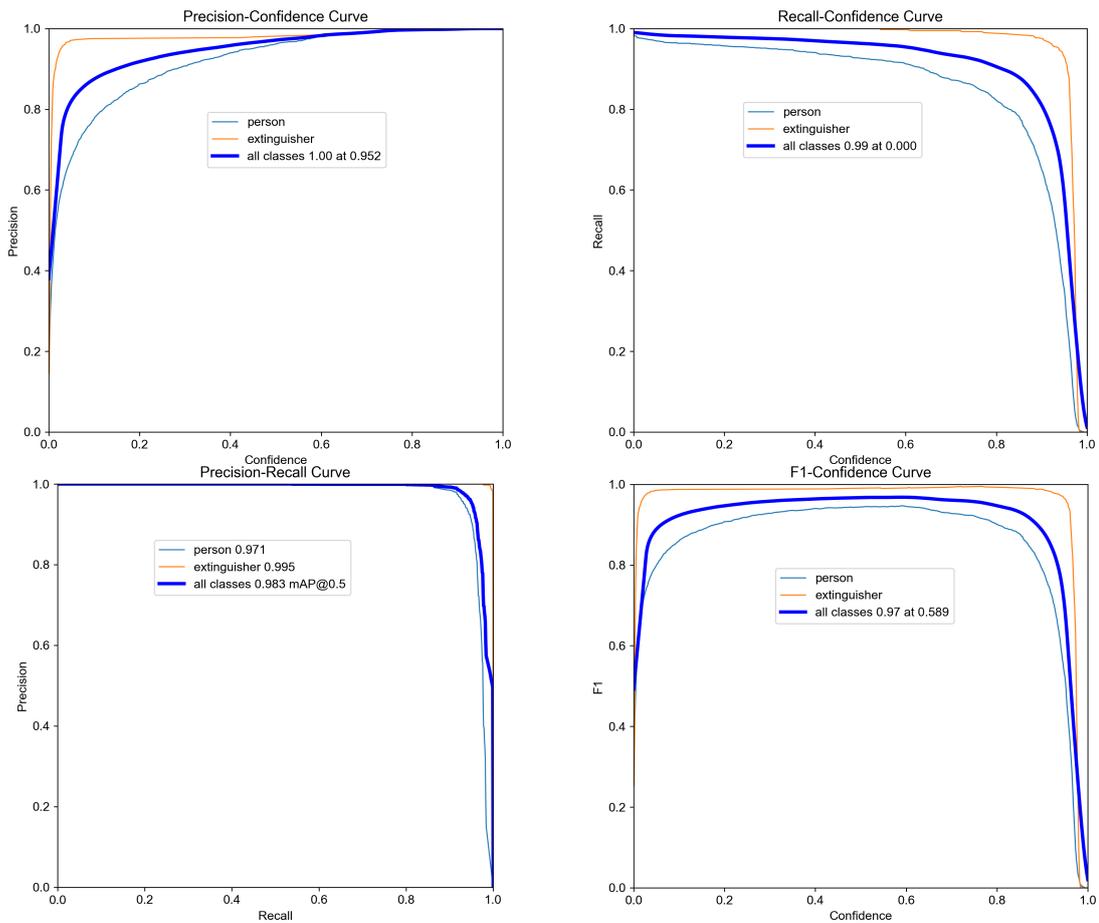


FIGURE 7.20 – Courbes de performances.

7.5.3 Expériences

Dans le but de tester le nouveau détecteur d'objets entraîné, nous mettons en place une expérience qui vise à présenter les différents cas auxquels le système peut être exposé. Durant le déplacement du piéton, certaines images contiennent uniquement des extincteurs, uniquement des humains, aucunes des ces deux entités ou une superposition des entités.

De plus, pour tester la robustesse du système, des faux positifs de deux types sont ajoutés : faux positifs physiques qui sont des images d'extincteurs imprimées collées à des endroits jugés problématiques dans l'espace physique. Ces endroits présentent des configurations : où la détection est plus compliquée à cause de la différence de luminosité, où l'extincteur est posé de manière à créer un faux positif cohérent qui permettrait d'effectuer l'appariement sur un amer qui n'est pas détectable, et où l'humain fait obstruction à l'extincteur dans le champs de vision.

Le but est de voir si le réseau de neurones bi-classes arrive à détecter les extincteurs et les humains, s'il y a une confusion entre les deux classes et à quelle échelle (grand, moyen petit) de détection cette confusion se produit le plus. Cette expérience est la composition de tests unitaires menés afin de valider le fonctionnement du détecteur.

7.5.3.1 Résultats

Les figures 7.21 présentent plusieurs faux positifs (FP) pour la classe humain. Les faux positifs existants sont occurrents tous à la petite échelle. Le détecteur d'objet a effectivement des difficultés à reconnaître les objets de petite taille dans l'image. De plus, ce sont ces cas là où la confusion de classe est la plus vraisemblable. On remarque que dans ces figures, plus le piéton se rapproche de l'extincteur, plus le réseau est confiant qu'il s'agit d'un extincteur que d'un humain.

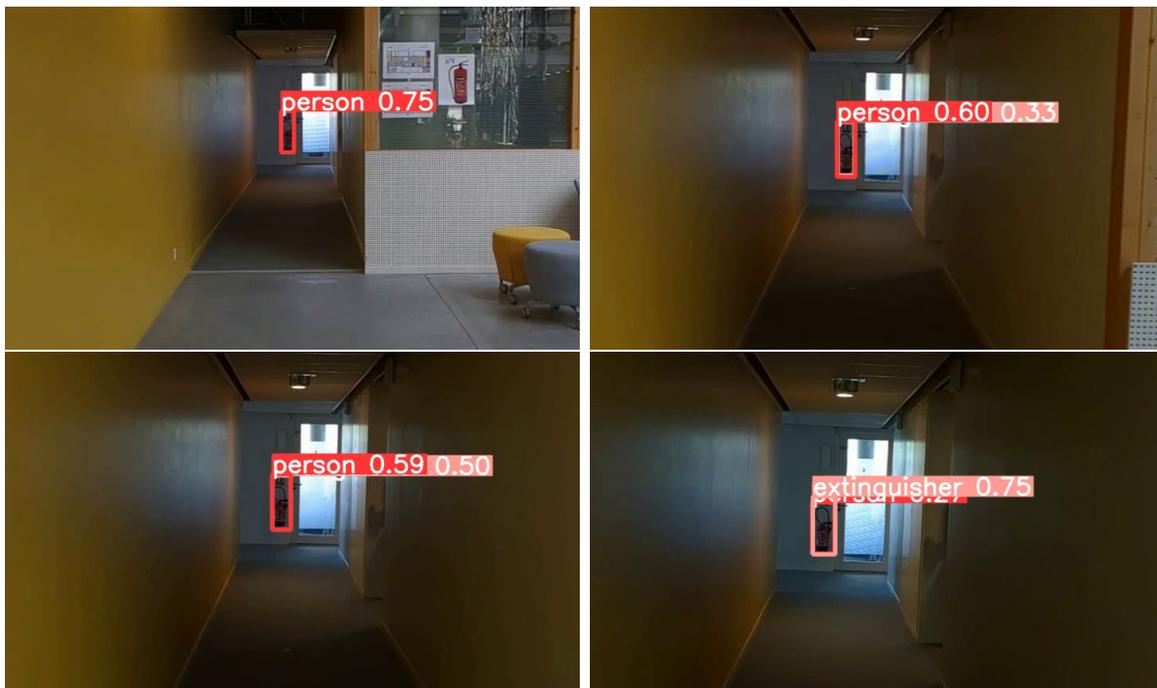


FIGURE 7.21 – Confusion entre extincteur et humain.

Dans la figure 7.22, à grande échelle, l'humain est distingué de l'extincteur avec des confiances élevées (>90%) même pour une luminosité relativement basse. Ceci provient du fait que dans le jeu d'entraînement, les extincteurs sont plus souvent représentés par des exemples plutôt proche de la caméra que éloignés. Ils tombent donc plus souvent dans la catégorie détection grande échelle.



FIGURE 7.22 – Humain et extincteur distingués.

Comme pour l'extincteur, les réflexions sont aussi problématiques pour la détection d'humain. Dans les figure 7.23, l'humain dans le visuel est détecté avec une confiance supérieure à 90%. Ses reflets le sont aussi à la même confiance. Mais, contrairement à l'extincteur, où la localisation permet d'exploiter un *a priori* sur la position tirée de la carte, l'humain ne possède pas un tel *a priori* et rien n'interdit la présence de deux humains à ces endroits.



FIGURE 7.23 – Réflexions présentes dans l'environnement.

Les figures 7.24 présentent plusieurs humains et les détections associées à plusieurs échelles. Le détecteur d'objet peut détecter correctement l'humain même quand une partie de celui-ci

est obstruée. Ceci est lié aux jeu de données COCO et à l'augmentation par mosaïque qui font que des exemples montrant l'humain incomplet sont présentés à YOLO durant l'entraînement.



FIGURE 7.24 – Détections de plusieurs humains à différentes échelles.

Enfin, le placement des faux positifs d'extincteurs imprimés permet de produire des fausses alarmes sur des extincteurs qui n'existent pas dans la carte mais sont présents dans l'image. La figure 7.25 montre un exemple où malgré la basse luminosité de la scène, le détecteur arrive à détecter cet extincteur imprimé.



FIGURE 7.25 – Détection sur un faux positif fabriqué.

7.6 Récapitulatif

Dans ce chapitre, nous avons exposé notre méthodologie permettant d'adapter le détecteur au problème de recalage visuel dans le cadre d'une localisation piétonne. Cette méthodologie explique le raisonnement et la procédure suivis pour le choix des amers à détecter. La discussion du choix de l'amer représente une partie importante de l'adaptation du modèle à la problématique. Si le cahier des charges du système de localisation venait à changer, le choix de l'amer peut ne pas se porter sur l'extincteur comme dans notre cas. L'extincteur a été choisi via une analyse du compromis entre présence, identifiabilité, exploitabilité, densité et standardisation.

La deuxième phase de notre méthodologie consiste à construire ou trouver un jeu de données ayant les exemples pertinents. Il faut valider l'intégrité des exemples inclus i.e. il peut y avoir un désaccord sur la validité d'un exemple dans le cas où deux extincteurs sont proches et sont étiquetés avec une seule étiquette. Le jeu de données d'images FireNet est dédié aux équipements anti incendie. Il a été épuré afin d'extraire les exemples pertinents d'extincteur pour entraîner YOLO dessus. Le nombre d'exemples dans ce jeu de données épuré est relativement faible, environ 200 images. Il a donc fallu réfléchir et proposer un stratagème permettant de compenser ce manque. L'adoption de l'augmentation des données est une première approche pour augmenter virtuellement le nombre d'exemples. Dans notre cas, une symétrie horizontale permet de doubler ce nombre. Ensuite, le *framework PyTorch* permet d'augmenter en ligne les images en mosaïque afin de rendre plus robuste l'entraînement selon les auteurs de YOLOv5 [Jocher et al., 2020].

La deuxième proposition pour que cet apprentissage soit faisable et robuste malgré le manque de données et d'entraîner YOLOv5 par transfert en prenant la version de ses auteurs entraînée sur le jeu de données généraliste MS COCO [Lin et al., 2014], puis, il faut spécialiser ce modèle pour la détection d'extincteur. Cette spécialisation passe par la congélation des couches convolutives et le ré-entraînement de la couche de classification qui produit la détection. Nous avons ensuite procédé à l'entraînement en suivant l'évolution des métriques présentées. Ces métriques montrent une convergence sans sur-apprentissage. L'apprentissage par transfert fait que le nombre d'exemples nécessaires pour la convergence du réseau de neurones est moins important qu'initialement prévu, d'environ mille fois. De plus grâce à l'approche de congélation des couches présentées dans ce chapitre, le temps d'entraînement par époque pour 400 images est plus court car les couches congelées n'ont aucune mise à jour des poids à effectuer.

Les performances d'un modèle basé sur des réseaux de neurones dépendent fortement de la corrélation entre les données d'entraînement et les données réelles acquises lors du déploiement. Des différences matérielles peuvent exister, par exemple si les caméras utilisées ont des paramètres optiques différents. De plus, des facteurs tels que la luminosité et l'exposition peuvent également affecter les performances, bien que les réseaux de neurones soient moins sensibles à ces phénomènes.

Il est important de noter qu'il peut y avoir des variations entre les modèles ou les versions d'extincteurs présents dans le jeu de données et ceux trouvés dans l'environnement réel, bien que les extincteurs soient standardisés. Cela entraîne une certaine variance intra-classe pour la classe "extincteur". Dans le jeu de données FireNet, les extincteurs à poudre et les extincteurs à eau/CO2 sont regroupés en une seule classe. Bien qu'il soit possible de les classifier séparément, cela nécessiterait un re-étiquetage manuel de tous les exemples, ce qui serait une

tâche coûteuse en termes de temps et de ressources. De plus, cela demanderait également un plus grand nombre d'exemples d'apprentissage pour apprendre à YOLO à distinguer les deux types d'extincteurs qui sont déjà très similaires.

Certains des faux positifs montraient une confusion entre un humain et un extincteur sur certaines images. L'explication la plus plausible à la confusion entre ces deux classes est donnée par l'existence d'antécédents d'entraînement de la classe humain dans le réseau de neurones. Cette classe qui possède une variance intra-classe élevée, est difficile à détecter dû à la variété de descripteurs de l'humain et de possibilités de les retrouver dans l'image ; et est la plus présente dans le jeu de données de départ COCO.

Il est donc nécessaire d'entraîner le réseau à détecter la classe humain afin de lui apprendre à la différencier de l'extincteur. Pour cela nous avons proposé de combiner les jeux de données épurés : FireNet et COCO pour que le premier apporte environ 400 exemples d'extincteur et le second apporte environ 1600 exemples d'humain. L'entraînement par transfert est effectué et ses résultats montrent la convergence de notre modèle personnalisé. Une deuxième campagne de tests est menée pour montrer les performances de ce nouveau entraînement. La confusion entre les deux classes n'est plus présente pour la détection à grande et moyenne échelles. Cependant, pour la petite échelle, la confusion est toujours présente mais à des niveaux de confiances relativement bas.

L'entraînement du détecteur pour reconnaître les humains représente l'étape initiale de l'objectif [H] car en détectant des piétons dans l'image, il est possible de les localiser par rapport au repère de la caméra en prenant en compte la projection au sol. Cette localisation permettrait un fonctionnement multi agents où chaque piéton équipé de ce système pourrait échanger avec un autre afin de fournir la position de celui-ci et l'aider à affiner son positionnement. Le système peut effectivement être distribué avec des interactions localisées toutes encadrées et contraintes par la carte du bâtiment.

7.7 Conclusion

Dans ce chapitre, notre méthodologie d'adaptation du détecteur d'objet YOLO , qui constitue une contribution de plus de ce manuscrit, a permis de l'entraîner afin de détecter l'amer que l'on a choisi : l'extincteur. Cette méthodologie est l'objectif [D] de ce manuscrit. Le but de détecter cet amer est de produire une observation à partir du signal extéroceptif pour recalculer visuellement le cap du piéton lors de la localisation. Elle permet de valider la brique de détection des amers. L'intégration de cette brique dans notre approche système est une autre de nos contributions. Les problèmes liés à l'usage des réseaux de neurones font aussi leur apparition après qu'une campagne de tests dans l'environnement a été menée. L'entraînement multi-classes a été abordé pour solutionner un cas limite présent dans la campagne de tests menée. C'est dans le chapitre suivant que le déploiement de notre détecteur est présenté. La mise en œuvre du recalage visuel sur les extincteurs et son incorporation au filtre particulaire y sont exposées. Une solution exploitant les contraintes géométriques tirées de la carte est déployée afin de palier les fausses alarmes.

Chapitre 8

Correction de trajectoire par observation de signaux extéroceptifs

Sommaire

8.1	Recalage visuel	160
8.1.1	Alignement temporel détection-pas	160
8.1.2	Construction de l'observation	161
8.1.3	Phase d'appariement	162
8.1.4	Mise à jour des particules	163
8.1.5	Expériences et résultats	165
8.1.6	Récapitulatif	169
8.2	Gestion des faux positifs	169
8.2.1	Applications des contraintes géométriques de la carte	169
8.2.2	Vote majoritaire et définitions des groupes	171
8.2.3	Expériences et résultats	172
8.2.4	Récapitulatif	178
8.3	Conclusion	179

Dans le chapitre 6, le recalage visuel est présenté comme une solution extéroceptive viable dans le cas où le piéton porte un téléphone et que les amers identifiés comme importants sont présents dans l’environnement. La détection de ces amers est présentée dans le chapitre 7 où l’on a constaté l’apparition de faux positifs qui doivent être gérée par le système. Ce chapitre, présente la mise en œuvre de ce recalage visuel dans notre système. À la fin du chapitre, notre système comme se résume à la figure 8.1. Ceci implique que le chapitre aborde l’exploitation de la détection pour la transformer en observation pour effectuer un recalage visuel dessus. Il présente aussi la validation de détection par contraintes géométriques tirées de la carte pour gérer les faux positifs que peut produire le détecteur.

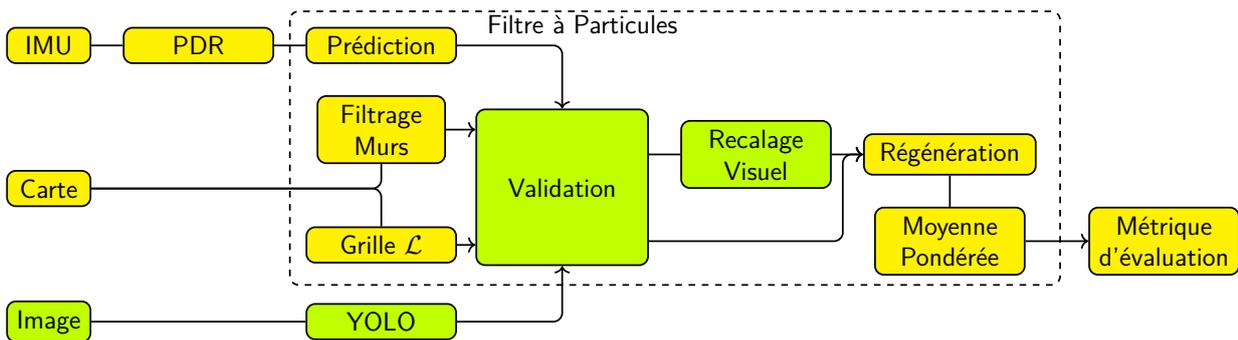


FIGURE 8.1 – Système proposé dans ce chapitre.

8.1 Recalage visuel

Le recalage visuel est le processus faisant appel à un signal extéroceptif de type image afin de soit corriger la position/cap du piéton, soit de modifier la vraisemblance de l’information durant la localisation. Plus formellement, il s’agit de mettre à jour les informations contenues dans le vecteur d’état représentant le piéton en fonction de l’information obtenue par une observation. Une observation doit être construite à partir d’une détection d’amer qui est produite par le détecteur d’objet YOLO présenté au chapitre 6. Dans ce chapitre, nous considérons que seul un amer peut être observé à la fois. Donc, il ne peut y avoir qu’une seule boîte englobante produite par YOLO sur une image. Cette hypothèse est tirée de l’analyse de la densité des amers présents sur la carte de l’étage dans lequel le piéton est.

8.1.1 Alignement temporel détection-pas

Dans le chapitre 3, le processus de détection de pas est présenté. Les événements produits par cette détection de pas cadence le traitement par le filtre à particules. Donc, si le système doit prendre en compte une information extéroceptive, elle doit être prête à l’emploi au moment où l’évènement de pas se produit. Dans le chronogramme de la figure 8.2, l’IMU fait une acquisition systématique à 50 Hz. La caméra fait aussi une acquisition systématique à 25 Hz.

Chaque image est fournie à l’entrée de YOLO afin qu’il infère les détections. Les pics rouges indiquent la présence d’une détection et leurs amplitudes indiquent la confiance associée à cette détection. La fenêtre d’observation est le temps minimum défini entre deux pas successifs ou 0,5 s (l’inverse de 2 Hz). Ceci est lié à notre hypothèse qui dicte que le cap ne change pas

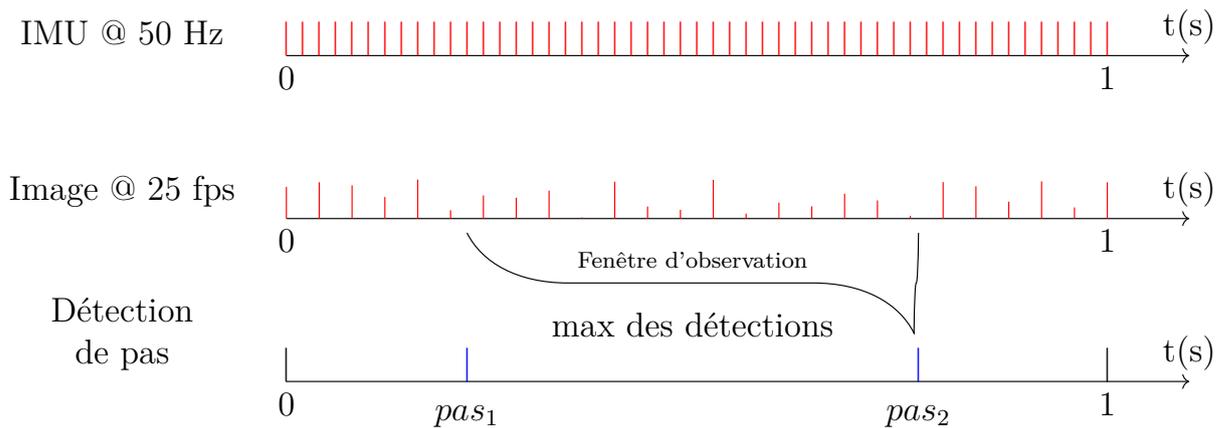


FIGURE 8.2 – Chronogramme des événements.

durant un pas. La détection ayant la plus grande confiance dans cette fenêtre d’observation est sélectionnée comme candidate pour en construire une observation. Ceci est équivalent à un calcul de maximum de vraisemblance pour la détection sur cette fenêtre.

8.1.2 Construction de l’observation

Une détection est sélectionnée comme candidate pour devenir une observation si la confiance que YOLO a en cette détection est supérieure à un seuil $conf_{ths}$. Le résultat de l’entraînement a montré que pour un seuil à 0,89 les vrais positifs étaient tous correctement détectés. De manière expérimentale, un seuil de 0,75 semble être satisfaisant.

Pour un amer L détecté, une observation O_t est construite par le filtre. O_t est défini comme :

$$O_t = \begin{bmatrix} [u, v]^T \\ [du, dv]^T \\ [x_g, y_g, z_g]^T \end{bmatrix} \quad (8.1)$$

Où :

- $[u, v]^T \in [0, 1] \in \mathbb{R}^2$ sont la position relative de la boîte englobante dans l’image.
- $[du, dv]^T \in [0, 1] \in \mathbb{R}^2$ sont les hauteur et largeur relatives de la boîte englobante dans l’image.
- $[x_g, y_g, z_g]^T \in \mathbb{R}^3$ sont la position de l’amer dans le SC global représenté sur la figure 8.3.

Quand O_t est instanciée, toutes ces composantes ne sont pas connues. Le processus de recalage visuel exploite les informations disponibles dans d’autres sources pour compléter celles-ci :

1. Le PF doit faire correspondre l’amer détecté parmi tous les amers possibles présents sur la carte dont les positions sont connues. Il s’agit d’un problème d’appariement qui requiert une procédure dédiée pour produire un couple (O_t, L) : une observation et un amer dans la carte.
2. Le PF doit aussi comparer l’information contenue dans chacune de ses particules à l’observation pour effectuer une correction du cap de celle-ci. Pour cela, une particule doit d’abord vérifier si l’amer est dans son champ de vision. Donc, il y a obligatoirement

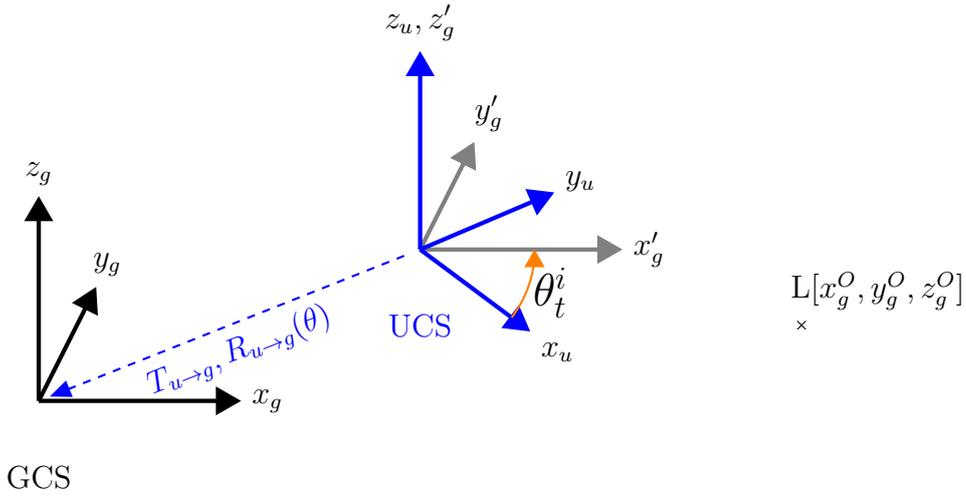


FIGURE 8.3 – Système de coordonnées de particule (UCS) par rapport au système de coordonnées global (GCS).

une prise en compte du modèle de la caméra et des contraintes géométriques imposées par la carte de l'étage.

Quatre grandeurs importantes sont introduites par notre méthode de recalage visuel :

1. Ω_t^O , **observation de amer** : l'angle dans lequel la caméra a observé L dans l'image.
2. $\tilde{\Omega}_t^i$, **observation de prédiction** : l'angle dans lequel la particule aurait vu L "si elle avait une caméra".
3. $\Delta\Omega_t^i$, **différence d'observation** : l'angle nécessaire pour que le cap de la particule puisse "voir" l'amer là où la caméra le voit (avec incertitude).
4. $\mathcal{L}(\Delta\Omega_t^i)$, **vraisemblance sémantique** : représente l'importance accordée à une particule en fonction de la différence d'observation.

La transformation des pixels à l'espace 3D est faite via le modèle de caméra sténopé qui est présenté en annexe C.1. Cette annexe présente le calculs des angles d'observation pour tout ce document.

8.1.3 Phase d'appariement

Avant de mettre à jour le vecteur d'état d'une particule, il faut déterminer quel amer la particule "croit avoir vu", c'est-à-dire quel amer L^j correspond à la particule P^i . Pour ce faire, l'algorithme 2 est une application définie comme :

$$f : (i, \Omega^O) \longrightarrow \tilde{j} \quad | \quad j \in \text{card}(\mathcal{L}) \quad (8.2)$$

Où \mathcal{L} est l'ensemble de tous les amers observables sur la carte d'étage. Cette application essaie effectivement de faire correspondre une particule d'indice i à un amer d'indice \tilde{j} . Il peut également renvoyer aucun lors qu'aucune correspondance n'est possible. Deux critères importants sont pris en compte : une particule ne peut évidemment pas voir un amer derrière un mur ; et, $\Delta\Omega$ ne doit pas être trop important (empiriquement, $< \pi/5$). L'amer d'indice \tilde{j}

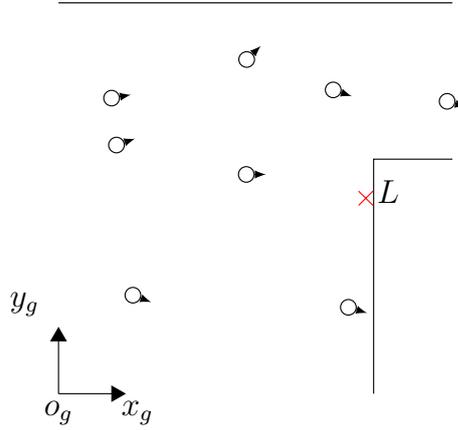


FIGURE 8.4 – Représentation de plusieurs particules devant s'apparier sur un amer L .

Algorithme 2 : Algorithme d'appariement observation-amer.

Entrées : P^i , Ω^O , \mathcal{L} , \mathcal{M}

Sortie : L_{match}

```

1  $L_{match} = \text{aucun}$ 
2  $\Delta\Omega_{min} = \infty$ 
3 pour  $L^j \in \mathcal{L}$  faire
4    $\tilde{\Omega}^{i \leftrightarrow j} = \text{calculer } \tilde{\Omega}_t^i (P^i, L^j)$  // en radians
5    $\Delta\Omega^{i \leftrightarrow j} = |\Omega^O - \tilde{\Omega}^{i \leftrightarrow j}|$ 
6   si  $\Delta\Omega^{i \leftrightarrow j} < \pi/5$  alors
7      $\Delta\Omega_{min} = \min(\Delta\Omega_{min}, \Delta\Omega^{i \leftrightarrow j})$ 
8     si  $\Delta\Omega_{min}$  a changé alors
9       si  $(P^i, L^j)$  ne traverse pas  $\mathcal{M}$  //  $\vec{PL}$  n'intersecte aucun mur
10      alors
11         $L_{match} = L^j$ 
12 Retourne :  $L_{match}$  // peut retourner aucun
```

qui satisfait ces deux critères et minimise le $\Delta\Omega$ est enregistré comme correspondance pour la particule i .

Chaque particule parmi N particules exécute ce processus pour essayer de trouver une correspondance. Les particules trouvant une correspondance utilisent l'amer apparié pour modifier leur poids. Celles qui ne trouvent pas de correspondance sont marquées comme invalides. La figure 8.5 montre un exemple pratique d'une détection avec une confiance élevée et la sortie d'appariement de amers. Il y a 2 extincteurs qui peuvent être observés sous le même angle par les particules. Notre système n'a pas de capacités de différenciation de profondeur, les deux sont des correspondances valides.

8.1.4 Mise à jour des particules

La figure C.1 montre la relation entre $\tilde{\Omega}_t^i$ et le système de coordonnées des particules (UCS). $o\vec{x}_u$ est orienté par le cap de la particule θ_t^i . Ainsi, la transformation du UCS au GCS



FIGURE 8.5 – Résultat du processus d'appariement observation - amer.

est exprimée par la translation $T_{u \rightarrow g}$ et par la rotation $R_{u \rightarrow g}$ définie comme :

$$R_{p \rightarrow g} = \begin{bmatrix} \cos(\theta_t^i) & -\sin(\theta_t^i) \\ \sin(\theta_t^i) & \cos(\theta_t^i) \end{bmatrix} \quad (8.3)$$

Dans ce cas il est possible de calculer $\Delta\Omega_t^i$ telle que :

$$\Delta\Omega_t^i = \tilde{\Omega}_t^i - \Omega_t^O \quad (8.4)$$

Et $\tilde{\Omega}_t^i$, d'après la figure 8.3 :

$$\tan \tilde{\Omega}_t^i = \frac{y_L}{x_L} \quad (8.5)$$

$$\tilde{\Omega}_t^i = \arctan \frac{y_L}{x_L} \quad (8.6)$$

Et Ω_t^O est donné par le modèle de sténopé présenté en annexe C.1. Enfin, en tirant parti des propriétés de représentation probabiliste PF, chaque particule voit son poids mis à jour en fonction de la différence d'observation. Cela signifie que le nouveau poids d'une particule \tilde{w}_t^i est défini comme :

$$\tilde{w}_t^i \propto w_t^i \times \mathcal{L}(\Delta\Omega_t^i) \quad (8.7)$$

Avec $\mathcal{L}(\Delta\Omega_t^i)$ la vraisemblance que de l'observation qui est définie par :

$$\mathcal{L}(\Delta\Omega_t^i) = \frac{1}{\sigma_{\Delta\Omega} \sqrt{2\pi}} \exp \left(-\frac{1}{2} \left(\frac{\Delta\Omega_t^i}{\sigma_{\Delta\Omega}} \right)^2 \right) \quad (8.8)$$

Cela signifie que nous supposons que l'erreur du modèle de sténopé est de distribution gaussienne centrée autour de Ω_O , l'angle auquel l'amer se trouve dans la caméra, avec un écart type, $\sigma_{\Delta\Omega}$ défini par l'équation (8.9) :

$$3\sigma_{\Delta\Omega} = \epsilon/2 \implies \sigma_{\Delta\Omega} = \pi/36 \text{ rad} \approx 5^\circ \quad (8.9)$$

Cet écart type est lié à l'angle d'ouverture du modèle de sténopé mesuré expérimentalement comme étant de $\epsilon/2 \approx \pm 15.5^\circ$. Plus la différence d'observation $\Delta\Omega_t^i$ est petite, plus la particule est représentative. Tous les poids sont normalisés avant la régénération présentée en chapitre 5. w_t^i est tiré de la grille de vraisemblance, elle aussi présentée en chapitre 5.

8.1.5 Expériences et résultats

La mise en place du recalage visuel dans un premier temps considère que toutes les détections sont des vrais positifs. La trajectoire choisie présente différentes configurations en intérieur. Certaines sont plus fermées comme des couloirs, d'autres sont semi-ouvertes ou ouvertes. Le phénomène de dérive, lié à l'usage des murs de la carte pour contraindre les particules du filtre particulaire, devrait être visible sur les trajectoires estimées. Nous configurons le PF avec exactement les mêmes paramètres que dans chapitre 5. Notre but sera de visualiser et de quantifier l'impact du recalage visuel, de comparer son déploiement avec son non déploiement. Nous souhaitons aussi connaître l'effet de la combinaison de la grille de vraisemblance qui est une de nos contributions avec le recalage visuel.



FIGURE 8.6 – Trajectoires pour quatre modes différents.

Dans la figure 8.6, pour une même séquence de données IMU et vidéo, quatre trajectoires sont représentées par quatre couleurs :

- En bleu, la trajectoire produite par le système présenté dans le chapitre 5 privé de la grille de vraisemblance.
- En vert, la grille de vraisemblance est activée pour le même système.
- En cyan, la trajectoire utilisant la caméra seulement pour effectuer le recalage.
- En rouge, la grille de vraisemblance et la caméra sont tous les deux actives.

Les trajectoires bleues et vertes montrent une dérive en cap très importante dans la zone ^①. Cette dérive introduit une erreur importante et elle n'est corrigée que par la présence du mur avant de rentrer dans le couloir. La zone ^② montre l'impact de la grille sur les trajectoires. La vidéo montre que le piéton est effectivement passé au sud du pilier. Pour la zone ^③ certaines particules mensongères survivent dans la courbe bleue ce qui fait qu'elles montrent un écart de position notable comparées aux trois autres. Ce qui montre que celle-ci n'est pas corrigée.

L'analyse qualitative de ces courbes montrent que l'implantation du recalage visuel a un impact important sur la qualité de la trajectoire estimée. La grille de vraisemblance a aussi pour effet de réduire ces erreurs à cause des transitoires entre espaces ouverts/semi-ouverts et fermés.

La figure 8.7 présente les distances euclidiennes et Mahalanobis moyennes sur 10 estimations Monte Carlo pour les mêmes données. Le mode rouge a clairement la meilleure valeur d'erreur euclidienne parmi les quatre modes. Sa distance de Mahalanobis indique que les ellipsoïdes d'incertitude sont plus resserrés. Le mode bleu montre une erreur et un écart-type plus important. Ceci montre un manque de robustesse quand le système ne déploie ni la grille de vraisemblance ni le recalage visuel.

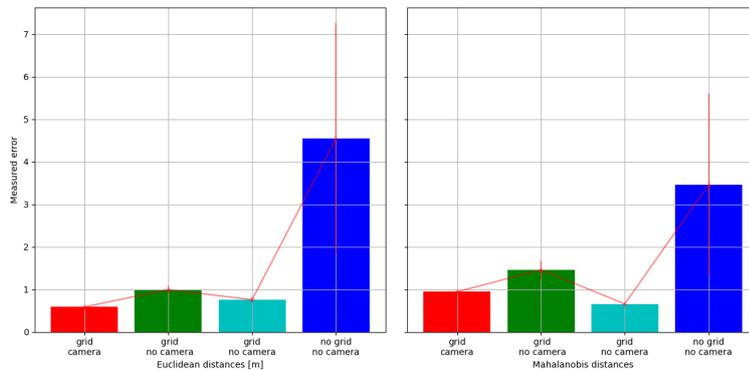


FIGURE 8.7 – Comparaison des erreurs euclidiennes et Mahalanobis moyennes.

Afin d'étendre la comparaison, le critère utilisé dans les conférences IPIN [Potortì, 2020], la CDF à 75% présentée au chapitre 4, est utilisé pour comparer ces quatre modes pour une réalisation. Dans la figure 8.8, les scores CDF classent du meilleur au pire : rouge, vert, bleu puis cyan. Ce critère approfondie l'analyse en incluant une notion de tolérance à l'erreur.

Il est ensuite question de comparer les intégrités des quatre modes afin de voir si un mode ne respecte pas l'exigence d'intégrité à trois écart-types d'erreur. Dans la figure 8.9, toutes les courbes rouge, verte, cyan incluent l'asymptote $y = 0$ dans le couloir d'incertitude. Seule la courbe bleue n'a pas cette asymptote emprisonnée entre $\pm 3\sigma$. Le mode bleu ne répond donc pas aux critères d'intégrité. Ces mêmes couloirs sont plus étroits pour les modes verts et rouge alors qu'ils le sont bien moins pour les modes bleu et cyan.

En isolant la courbe rouge de la figure 8.9, on obtient la figure 8.10 pour différents σ afin d'examiner les limites de cette intégrité. Le critère sur l'intégrité à deux et trois σ est bien

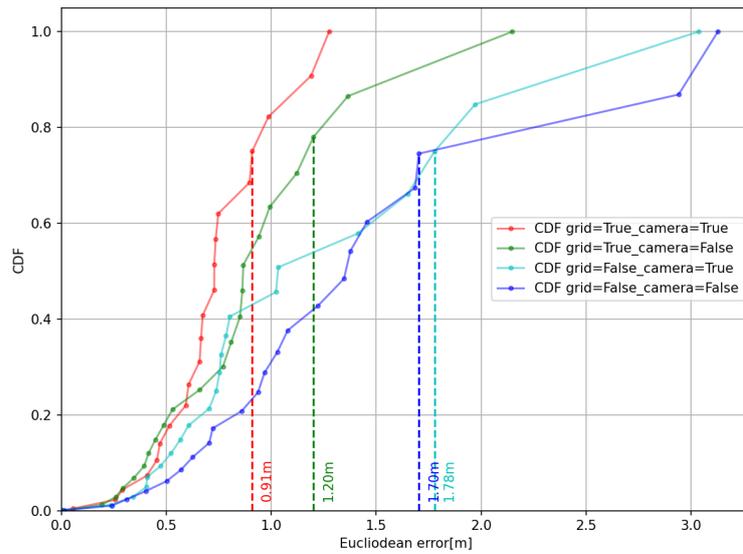


FIGURE 8.8 – Comparaison des scores CDF pour les quatre modes.

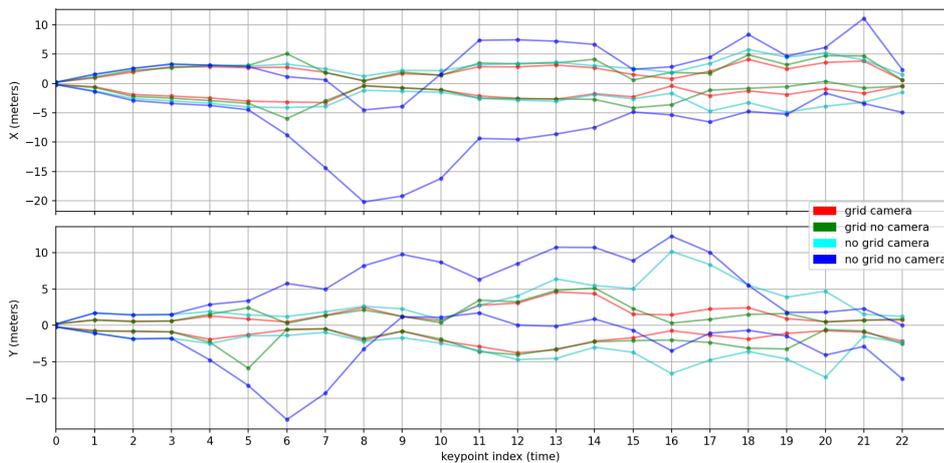


FIGURE 8.9 – Comparaison des critères d'intégrité pour les quatre modes.

respecté. Alors qu'à un σ , il n'est pas respecté au point 14 et 16.

La figure 8.11 présente le cas où le recalage visuel est toujours actif et que la grille de vraisemblance est une fois activée, une fois désactivée. Le but est d'établir si pour une trajectoire jugée difficile, le système garde sa robustesse face à une mésestimation de ce gain avec cette fois-ci l'ajout du recalage visuel. L'impact de notre grille vraisemblance est remarquable car sur 10 trajectoires pour les mêmes données, aucune n'échoue quand la grille est active. Alors que sans cette grille, les trajectoires échouent plus souvent.

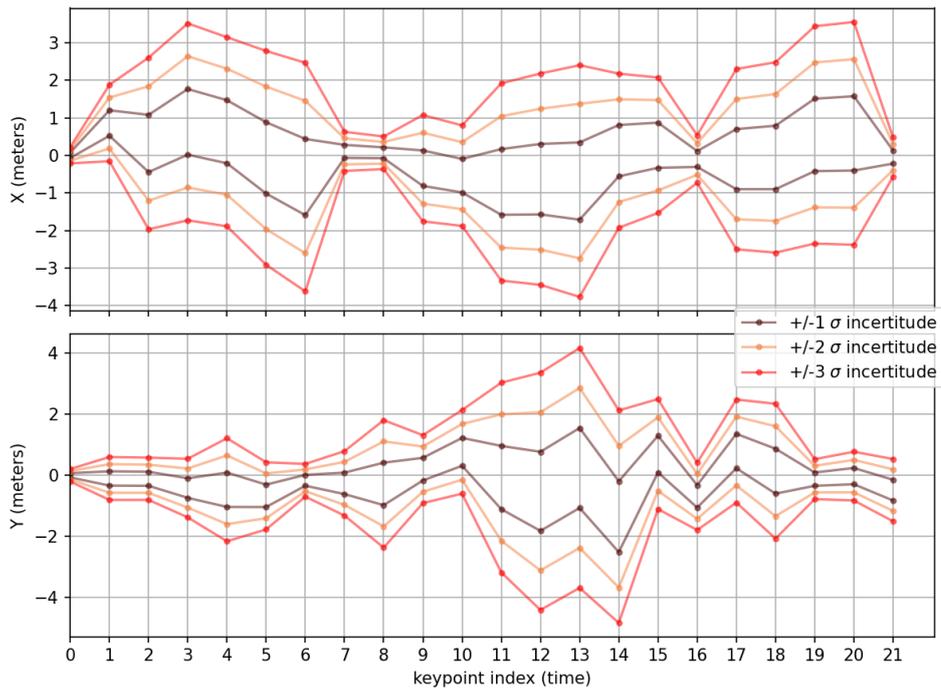


FIGURE 8.10 – Comparaison des critères d'intégrité pour le mode rouge pour différentes σ .

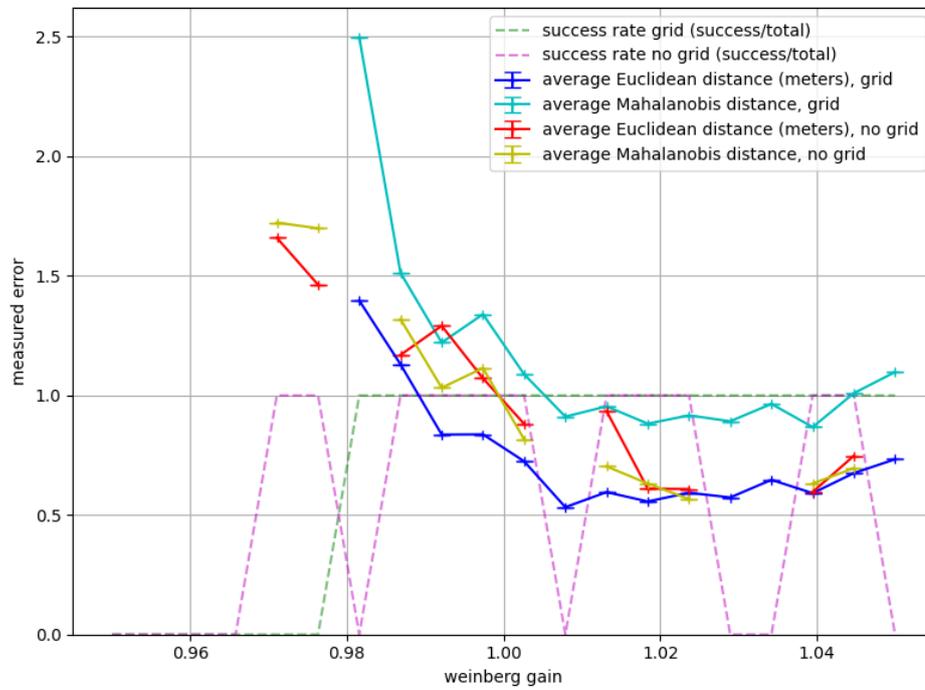


FIGURE 8.11 – Comparaison de la présence de grille sur le gain de *Weinberg* avec recalage visuel toujours actif.

8.1.6 Récapitulatif

Les résultats présentés ci-dessus montrent le recalage visuel dans le cas où le détecteur ne fournit que des vrais positifs au filtre particulaire. Le recalage visuel permet de corriger effectivement la trajectoire en modifiant les poids des particules en fonction de leur écart à l'observation. La phase d'appariement prend en compte les contraintes tirées de la carte pour permettre à chaque particule de choisir l'amer sur lequel elle se recalcule.

Les trajectoires présentées ont des passages ayant différentes difficultés : des couloirs, des halls et des espaces semi-ouverts. Le phénomène de dérive dû à l'absence de murs ou leur présence que d'un seul côté du piéton est corrigé en observant un amer. Ceci a un impact sur les distances euclidiennes car les particules ayant une différence d'observation importante et/ou ne pouvant pas s'apparier sont éliminées. Les performances obtenues satisfont les objectifs [A, B & C] du manuscrit.

Les résultats montrent aussi l'impact de la combinaison de la grille de vraisemblance avec ce recalage. Le système est plus performant dans le cas où les deux sont combinés contre les cas où chacun est le seul actif.

Toutefois, dans ces premières sections de chapitre, nous avons considéré que le détecteur d'objets produisait aucun faux positifs. Ce qui n'est pas le cas dans le cas pratique où des fausses alarmes peuvent être produites. La gestion de ces fausses alarmes est présentée ci-dessous.

8.2 Gestion des faux positifs

8.2.1 Applications des contraintes géométriques de la carte

Nous proposons d'utiliser la carte du bâtiment afin d'extraire des contraintes géométriques et spatiales permettant d'indiquer à quel point la détection en cours est valide par rapport à une observation O_t et à l'état de la particule P_t^i en cours de traitement. Plusieurs contraintes tirées de la carte sont étudiées et participent au calcul de la vraisemblance de la décision vrai ou faux positif de cette détection visuelle.

8.2.1.1 Vraisemblance par la largeur

À partir de la largeur de la détection, du , on calcule les bords gauche et droit de la boîte dans l'image et les angles, Ω_g et Ω_d correspondant à ceux-ci via le modèle sténopé par l'équation (8.10). La figure 8.12 donne une schématisation de ce processus.

$$u_{g,d} = u - du/2 \rightarrow \Omega_{g,d} = -\arctan \frac{u_{g,d} - C_u}{F_u} \quad (8.10)$$

On peut alors de chaque particule i lancer deux rayons : gauche et droit, r_g^i et r_d^i , dans la carte depuis sa position et de calculer les collisions de ces rayons avec les murs, K_g^i et K_d^i . Un point de collision P_g^i ou P_d^i est défini par l'équation (8.11) où . On se base effectivement sur l'*a priori* que les extincteurs sont toujours sur un support qui est vissé au mur à la verticale.

$$P_{g,d}^i = \begin{bmatrix} x_{g,d}^i \\ y_{g,d}^i \\ id_{g,d}^i \end{bmatrix} = \begin{bmatrix} R \cos(\theta^i + \Omega_{g,d}) \\ R \sin(\theta^i + \Omega_{g,d}) \\ \text{carte}(x_{g,d}^i, y_{g,d}^i) \end{bmatrix} \quad (8.11)$$

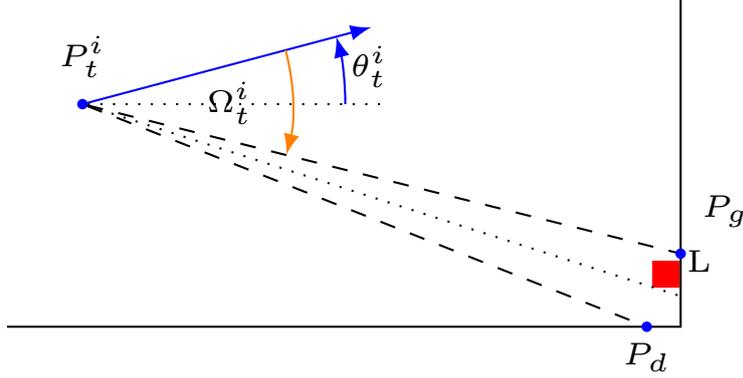


FIGURE 8.12 – Visualisation de la validation par largeur.

Où id est l'identifiant du mur auquel le point de collision appartient. R est un rayon de longueur infinie (dans la pratique, 200 m). On obtient ainsi 2 points de collision. Mais, il existe des cas limites qui complexifient cette tâche :

- Si $id_g == id_d$, les deux points appartiennent au même mur.
- Sinon :
 - + Sélectionner le point de collision K' le plus éloigné de la particule.
 - + Calculer l'intersection entre le mur d'identifiant id et le vecteur $\overrightarrow{P_t^i K'}$ comme le nouveau point de collision.

On peut alors calculer la largeur L_e du potentiel amer placé contre le mur telle que :

$$L_e^i = \sqrt{(x_g^i - x_d^i)^2 + (y_g^i - y_d^i)^2} \quad (8.12)$$

Cette largeur donne un premier indice sur la vraisemblance de la détection. Les extincteurs ont une largeur physique fixe et connue. La projection sur les murs de la largeur de la détection est sujette à l'angle d'incidence du rayon lancé avec le mur. Il est possible donc d'avoir des largeurs projetées plus importantes que les largeurs physiques mais qui sont du même ordre de grandeurs.

La vraisemblance de la validité de la détection par sa largeur, \mathcal{L}_L , peut être calculée via l'équation (8.13), où L_g est la largeur physique d'un extincteur standard. Cette vraisemblance est présumée gaussienne de centre L_g et d'écart-type σ_g .

$$\mathcal{L}_L^i \sim \mathcal{N}(|L_g - L_e^i|, \sigma_g) \quad (8.13)$$

8.2.1.2 Vraisemblance par la hauteur

Il est possible d'exploiter le fait qu'un objet dans la caméra occupe un espace pixel qui est en fonction de ses dimensions physiques et sa distance par rapport à l'obturateur. Afin d'exploiter ces deux aspects, certains *a priori* sont requis :

1. L'objet en question est de dimensions connues et constantes. On s'intéresse spécifiquement à la hauteur de l'extincteur qui est entre 74 et 78 cm.
2. Un des axes de rotation de cet objet doit être contraint. Dans le cas de l'extincteur, le fait qu'il soit toujours attaché au mur fait qu'il apparaît toujours à la verticale sur l'image.

3. L'extincteur et la caméra sont à hauteur constante du sol même si inconnues. Le positionnement d'un extincteur sur un mur est permanent durant toute sa durée de vie. La position de la caméra est quasi-fixe car elle rejoint notre première hypothèse sur le fait que le piéton essaie de garder le téléphone dans la même pose dans sa main.

Alors, le calcul de \mathcal{L}_H procède par :

1. Lancer un rayon dans l'angle $\theta^i + \hat{\Omega}^i$ i.e l'angle du cap dans lequel une particule prédit l'observation de l'amer potentiellement détecté.
2. Calculer la distance entre la particule et le point d'impact avec le mur, d_i .
3. Calculer \mathcal{L}_H qui est assimilée à une gaussienne de moyenne $f(du)$ et d'écart-type $\sigma_{g(du)}$ selon l'équation (8.14).

$$\mathcal{L}_H^i \sim \mathcal{N}(|d_i - f(du)|, \sigma_{g(du)}) \quad (8.14)$$

L'obtention de la fonction f est présentée dans l'annexe C.1.

8.2.2 Vote majoritaire et définitions des groupes

La validation multi-hypothèse d'une détection est schématisée dans la figure 8.13. Les processus de validations ont pour fonction de permettre de trier les particules en trois groupes, qui sont :

- Validant (V) : ce groupe de particules croient que la détection est correcte et utilise l'observation afin d'effectuer le recalage visuel.
- Non Validant (NV) : ce groupe ignore la détection et sa mise à jour fait décroître les poids des particules ou non.
- Invalides (I) : ce groupe a enfreint une contrainte de déplacement. Il est éliminé et doit être régénéré suivant le processus de régénération du chapitre 5 [Ghaoui et al., 2022].

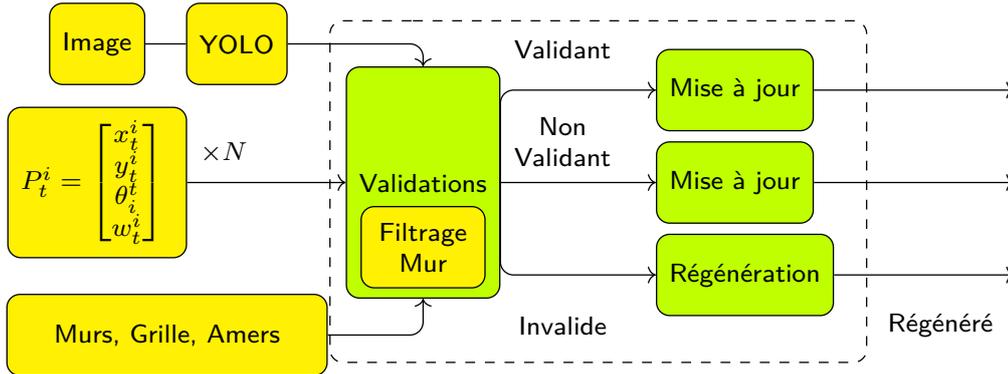


FIGURE 8.13 – Schéma bloc de l'intégration du processus de validation.

Ces 3 groupes existent jusqu'à la prochaine détection de pas où ils sont mélangés. En exploitant le poids d'une particule w_t^i et les vraisemblances introduites par les équations (8.13) et (8.14), on peut calculer \mathcal{L}_d^i la vraisemblance de la détection d selon la particule i via équation (8.15).

$$\mathcal{L}_d^i \sim w_t^i \times \mathcal{L}_L \times \mathcal{L}_H \quad (8.15)$$

Enfin, une particule i est dans le groupe validant si $\mathcal{L}_d^i > \lambda$. Où, λ le seuil de validation. Sinon elle est dans le groupe non validant. La valeur de λ est $10^{-4} = (N \times 100)^{-1}$, soit 10 fois le seuil d'élimination d'une particule.

Enfin, une dernière règle de validation est mise en place. Si toutes les particules sont dans le groupes invalide, l'observation est annulée et le système revient à l'étape de prédiction du filtre particulaire. Cette règle prend en compte le pire cas possible et assure que la localisation n'échoue pas pour un faux positif fatal.

8.2.3 Expériences et résultats

8.2.3.1 Effet du traitement différencié

La figure 8.14 est une visualisation des particules à un instant donné, chacune des 200 particules en bleu représentant une trajectoire possible dans le plan 2D du bâtiment. On peut constater que les particules se dispersent effectivement dans l'espace semi-ouvert mais avec une densité plus importante sur la droite. Le poids des particules à droite est aussi renforcé d'où la présence de pics magenta plus élevés sur celles-ci. Ces particules sont donc du groupe validant car elles ont été recalées visuellement sur un extincteur.

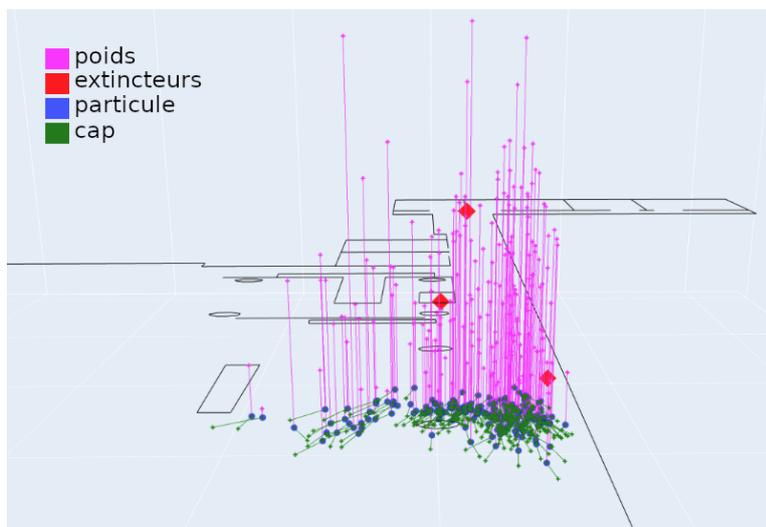


FIGURE 8.14 – Visualisation 3D des particules.

La figure 8.15 montre l'évolution du nuage à quelques instants où il y a eu une détection. Les particules en bleu sont du groupe validant. Les particules en rouge sont du groupe non validant. Le groupe invalide n'est pas affiché pour faciliter la lecture. Le groupe rouge est souvent présent contre le mur car il lui est impossible de valider la présence de l'amer parce que le lancé de rayon atterrit sur un obstacle. Quand une particule rouge n'est pas proche des murs, cela implique que son poids était faible.

La figure 8.16 montre la répartition des 2 groupes bleu et rouge pour différents seuils de validation λ . λ peut être paramétré pour définir une confiance plus ou moins élevée dans les détections qu'il reçoit. La figure montre comment le seuil de validation de la vraisemblance de la détection fait évoluer les 2 groupes.

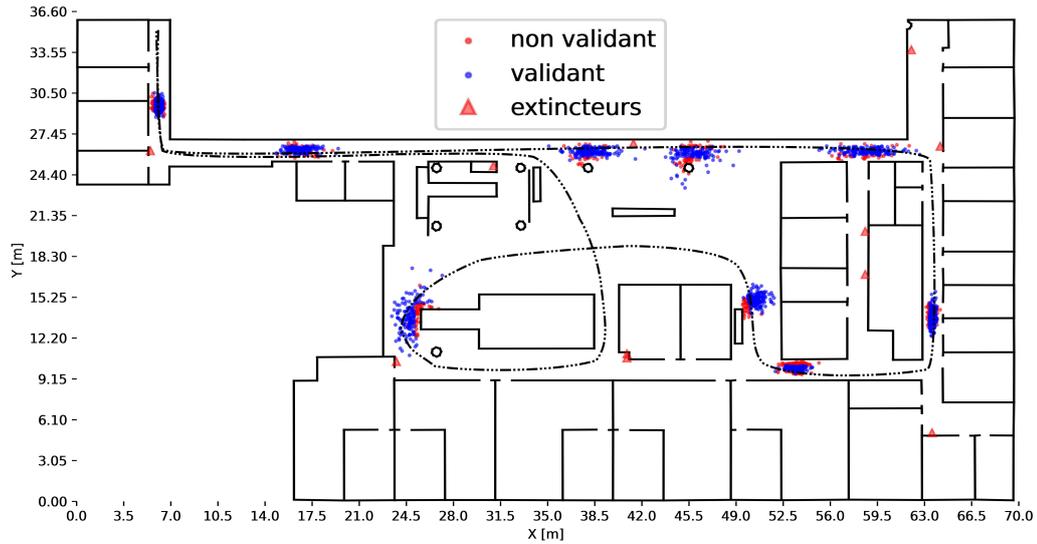


FIGURE 8.15 – Regroupement des particules sur une trajectoire.

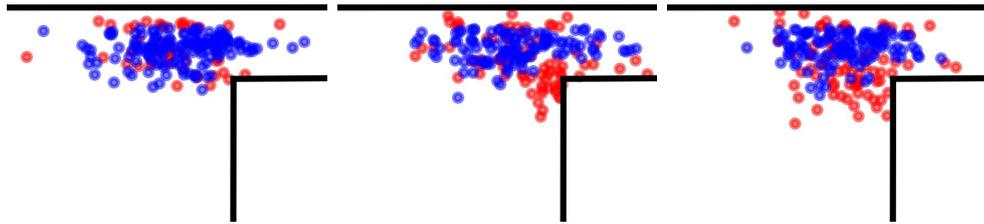


FIGURE 8.16 – Regroupement des particules à un pas donné pour λ , 2λ et 3λ .

Ceci fait de λ un paramètre de sensibilité et pourra donc servir à décrire un comportement plus évolué dans l'avenir.

Les particules du groupe I sont éliminées et régénérées. Les particules du groupe V sont toujours visuellement recalées. Par contre, nous comparons 2 traitements sur NV. Ce groupe ignore la détection et sa mise à jour laisse le poids de ces particules inchangé ou l'annule. Dans le premier cas, elles continuent sans recalage visuel. Dans le second cas, le poids est annulé et les particules sont régénérées comme pour I.

La figure 8.17 montre une erreur euclidienne moyenne plus élevée pour le traitement annulé que pour le traitement inchangé. Les faux positifs présents dans l'expérience mettent en faute le groupe validant qui se recalce dessus. L'information des particules qui n'avaient pas validé un faux positif est perdue selon le traitement annulé. Ce qui fait que le système de localisation est induit en erreur par leur absence.

Quant à la distance de Mahalanobis, le traitement annulé possède une plus grande distance car la distribution de ses particules possède une variance plus faible. Donc, le traitement annulé ne représente pas aussi bien l'espace des hypothèses que le traitement inchangé.

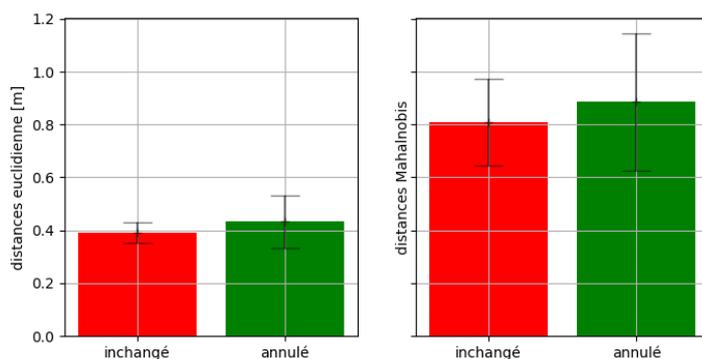


FIGURE 8.17 – Comparatif entre deux traitements pour le groupe non validant.

8.2.3.2 Effet des faux positifs sur les trajectoires

Dans ces expériences, le système est soumis à des scénarios de localisation variés en difficulté. Quatre trajectoires sont parcourues dans notre environnement intérieur. Le détecteur utilisé est celui résultant de l'entraînement bi-classe (extincteur et humain) du chapitre 7. Dans certaines de ces expériences des faux positifs d'extincteur sont mis en place sur la trajectoire de manière à produire une détection comme vu dans le chapitre 7. Mais, ceux-ci n'existent pas. D'autre part, des détections peuvent être ajoutées de manière à fauter le détecteur i.e. produire artificiellement un faux positif en créant une boîte englobante virtuelle qui l'incarne. Enfin, certaines de ces trajectoires ont des humains qui parfois se présentent entre la caméra et l'extincteur.

- Trajectoire 1 : celle-ci ne contient ni humains, ni de faux positifs physiques, ni de faux positifs virtuels. Seuls les faux positifs que le détecteur peut produire sont inclus. Celle-ci possède quelques espaces semi ouverts où le piéton a parcouru environ 10 m.
- Trajectoire 2 : ajout des faux positifs virtuels ainsi qu'une plus grande distance parcourue en espace ouvert (environ 20 m).
- Trajectoire 3 : ajout de faux positifs physiques et des humains dans la trajectoire, environ 20 m en espace ouvert.
- Trajectoire 4 : ajout de faux positifs physiques et virtuels ; des humains avec une trajectoire dans des espaces ouverts plus importante d'environ 50 m.
- Trajectoire 5 : même configuration que la trajectoire 4 avec une distance plus grande à parcourir.

Les figures 8.18 à 8.21 présentent chacune une réalisation des trajectoires demandées au piéton. Dans chacune, l'ordre de passage par les points de mesure en cyan est défini à partir du point 0. Les distances euclidiennes E et de Mahalanobis M moyennes sont récapitulées dans le tableau 8.1. Pour la figure 8.18, l'estimation de la trajectoire paraît cohérente avec le mouvement demandé même si elle indique que le piéton n'est pas exactement retourné au point de départ. Le demi tour en (63 m, 32 m) est bien effectué. La disposition des amers permet de maintenir la cohérence la trajectoire et d'assurer les métriques suivantes $E=0.49$ m et $M=0.90$. La dérive causée par la présence de la grille de la vraisemblance et des murs autour des points 1 et 7 peut être constatée. Les faux positifs produits par le détecteur peuvent manquer de cohérence dans le cas où ils ne sont pas appareillables à un amer présent sur la carte. Dans ce cas, les particules décident que la détection est invalide à cause de sa vraisemblance très

faible.

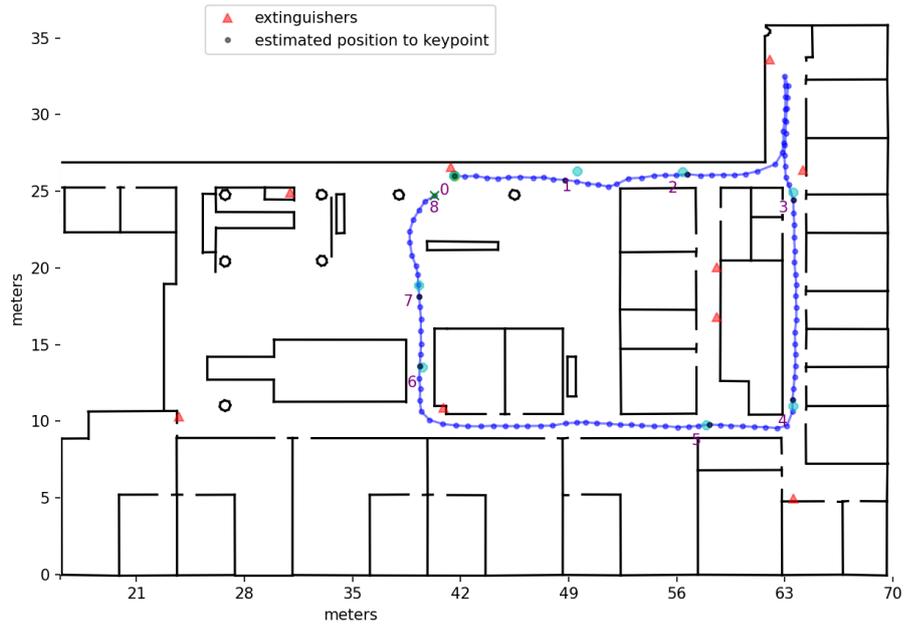


FIGURE 8.18 – Trajectoire 1.

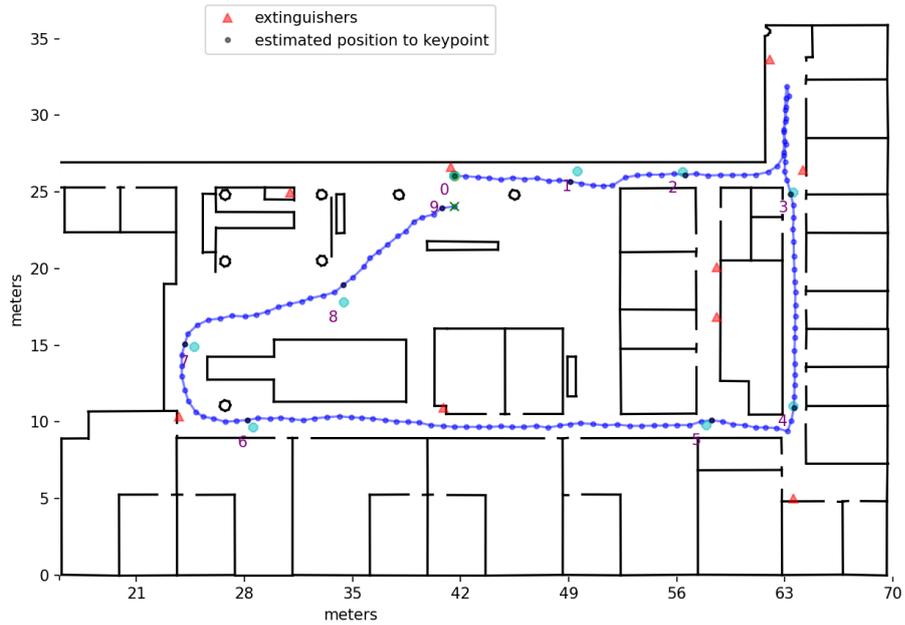


FIGURE 8.19 – Trajectoire 2.

Dans la figure 8.19, la trajectoire 2 présente des erreurs $E = 0.48$ m et $M=0.87$ similaires à la trajectoire 1. Le passage présentant le plus de difficultés est entre les points 7, 8 et 9 qui représentent environ 18 m d'espace ouvert sans un amer disponible pour fournir l'information extéroceptive. Ceci laisse le système en navigation proprioceptive qui dépend d'à quel point les estimations antérieures sont précises. C'est notamment sur le point 8 que l'erreur euclidienne est la plus importante. Les faux positifs virtuels sont gérés grâce à l'incohérence de leur

positionnement dans la carte par rapport à leur placement dans l'image. Ce qui résulte en une vraisemblance d'apparition faible accordée à cette détection et donc le groupe de particules non-validant contient moins de particules.

Dans la figure 8.20, le système estime la trajectoire du piéton et obtient les distances moyennes $E=0.63$ m et $M = 1.02$. La présence de faux positifs physiques à des endroits où le système peut produire une confusion est visible entre les points 6 et 7. L'image d'un extincteur collée au pilier permet de générer une détection qui au-delà d'avoir une confiance élevée et aussi cohérente car elle passe le processus de validation qui lui accorde une vraisemblance très importante. De plus la présence du véritable extincteur en (23 m, 10 m) fait que le groupe validant a donc un nombre important de particules qui vont se recaler sur ce faux positif physique. Cela renforce alors les particules sur la droite quand celles sur la gauche du piéton ne peuvent s'appareiller avec l'amer sur la carte. Cela produit un biais sur la moyenne pondérée qui est visible. Le système réussit quand même à continuer d'estimer la trajectoire car certaines particules ont invalidé la détection et continuent à se propager. Au moment où le virage à 90° arrive, la carte filtre les particules mensongère et la grille fortifie les particules survivantes. Ceci rapproche la représentation de l'espace des possibles de la réalité du piéton, d'où le passage par le point 7. La trajectoire du point 8 au 10 est une distance parcourue tout droit en espace ouvert. Dû à la symétrie locale de l'environnement (i.e. des murs des deux côtés), la position moyenne est relativement correcte autour de ces points.

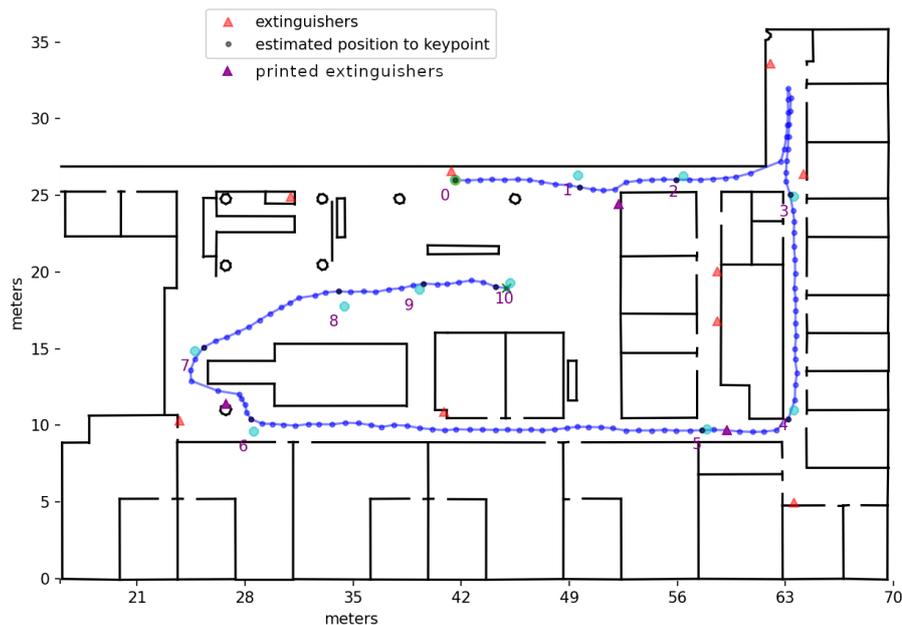


FIGURE 8.20 – Trajectoire 3.

Dans la figure 8.21, le système estime la trajectoire du piéton avec $E=0.77$ m et $M = 0.89$. La trajectoire gagne en complexité à cause des passages par un espace semi ouvert entre le point 0 et 2 qui permet la dérive des particules sur le côté droit du piéton jusqu'à être corrigée par les murs de la carte avant le point 3. Le passage aux points 8, 9, 10 (confondu avec 1) et 11 est plus complexe car il reprend la configuration locale de la trajectoire 3 autour du point 7 mais, au lieu d'avoir une trajectoire aux murs symétriques entre les points 8 et 11, le piéton passe près de zones semi-ouvertes (murs d'un côté et pilier) ce qui rend la localisation

plus difficile à cause du manque d’amers en visibilité directe et détectable. L’amer au point 1 représente une petite cible quand le piéton est au point 9. Ce qui rend sa détection plus difficile et donc la vraisemblance accordée à l’observation faible. Ce qui implique que peu de particules se recalent sur cet amer quand il est détecté à cette distance là. La trajectoire entre le point 10 et 11 ne possède aucun amer en vue pour effectuer un recalage visuel. La nécessité d’amers dans cette zone est plus importante. Cela montre que malgré le choix des extincteurs comme amers qui semble avoir une densité de répartition satisfaisante, il existe encore un besoin supplémentaire d’amers dans ce cas de figure.

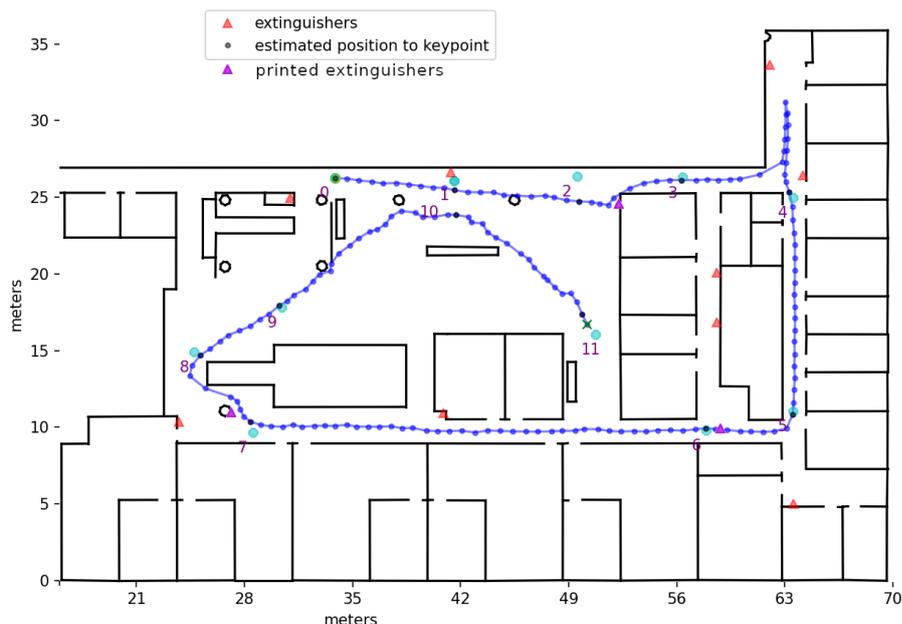


FIGURE 8.21 – Trajectoire 4.

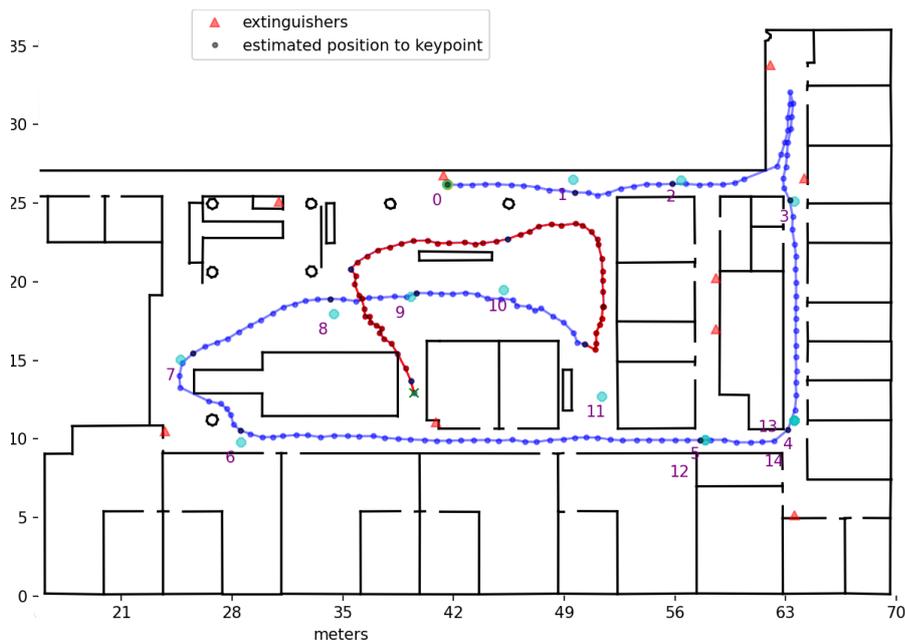


FIGURE 8.22 – Trajectoire 5.

Dans la figure 8.22, le système estime la trajectoire du piéton avec $E=5.60$ m et $M = 6.60$. Cette trajectoire reproduit le mouvement de la trajectoire 4 mais va plus loin à travers les points de passages 12, 13 et 14. Sauf que l'espace des possible avant le point de passage 11 ne peut pas résoudre l'ambiguïté sur l'estimation. Il est en effet possible que le piéton soit allé tout droit puis a fait un virage vers sa gauche ; Ou bien il est remonter. L'absence d'amers autour de (50 m;9,5 m) fait que cette ambiguïté ne peut être correctement géré par la navigation proprioceptive, d'où la trajectoire rouge résultant sur la figure 8.22. Cette trajectoire rouge correspond au passage par les points 11, 12, 13 et 14, avec une correction de la part de la carte et de la grille de vraisemblance, mais avec une rotation.

Trajectoire	Erreur euclidienne moyenne [m] E	Distance de Mahalanobis moyenne M
1 - figure 8.18	0.49	0.90
2 - figure 8.19	0.48	0.87
3 - figure 8.20	0.63	1.02
4 - figure 8.21	0.77	0.89
5 - figure 8.22	5.60	6.60

TABLE 8.1 – Distances moyennes pour les cinq trajectoires.

8.2.4 Récapitulatif

Trois groupes de particules gèrent les faux positifs et différencient les traitements selon leur croyance en la présence de l'amer. Cette approche permet d'utiliser le paramètre λ décrivant la sensibilité du système à ces détections.

Les particules calculent leur croyance de l'amer en exploitant les informations contenues dans la carte. En tenant comptes des *a priori* sur l'amer et l'information apportée par l'observation, il est possible de déduire si une particule croit ou non que l'amer peut être présent quand la détection est produite.

Les particules croyant la présence de l'amer sont mises dans le groupe validant. Ce groupe est soumis au processus de recalage présenté dans ce chapitre. Il voit les poids de ses particules modifiés par les vraisemblances de la présence de l'amer. Ceci lui permet de corriger le cap du piéton dans ce cas.

Nous avons étudié l'effet de deux traitements pour le groupe non validant. Dans le premier, le groupe est inchangé. Les particules ne croyant pas que l'observation est valide l'ignore et aucune composante de leur vecteur d'état n'est modifiée. Le comparatif montre que le traitement inchangé est sensiblement meilleur en terme de précision de localisation avec une incertitude plus faible. Cette gestion des faux positifs répond à l'objectif [E] du manuscrit.

Dans un second temps, une série d'expériences a été conduite pour montrer la capacité de notre système à gérer les faux positifs de trois types : FP propres au réseau de neurones ; FP physiques et FP virtuels. Les expériences montrent une dégradation notable des métriques et que le système garde une trajectoire cohérente malgré les difficultés de celles-ci et son exposition à ces faux positifs. La difficulté de ces trajectoires réside dans les différentes configurations présentes dans l'espace intérieur. Certains passages présentent un manque d'amers qui permettrait au système de se recalculer visuellement. Une évaluation quantitative de l'impact des faux positifs groupés par type devra être effectuée afin de permettre de comprendre l'évolution des performances du système quand il est exposé à un environnement défavorable. Dans le cas

où il n’y a pas assez d’amers, le système peut effectivement échouer comme montré dans la figure 8.22. Dans ces cas l’erreur augmente de manière très conséquente.

8.3 Conclusion

Dans ce chapitre, la mise en œuvre du recalage visuel est présentée. Ce recalage passe par plusieurs procédures : la première vise à déterminer l’instant durant un pas pour sélectionner une détection parmi un ensemble ; la seconde construit l’observation à partir de la détection sélectionnée en définissant les repères et le modèle sténopé nécessaires pour obtenir l’angle d’observation dans lequel l’amer réside ; une phase d’appariement est déployée afin d’associer l’observation à un amer existant sur la carte du bâtiment dont la position est connue. La mise à jour des particules se fait en fonction de l’écart à l’observation où une particule voit son poids renforcé en fonction de sa minimisation de cet écart. Le recalage visuel montre une réduction importante de l’erreur de localisation et l’incertitude associée à celle-ci, ce qui correspond à l’objectif [A,B & C]. Les expériences ont aussi montré que le recalage est nécessaire dans des scénarios plus compliqués où les hypothèses du filtre particulaire en localisation proprioceptive ne peuvent pas à elles seules déterminer correctement la position du piéton, d’où la nécessité d’un signal extéroceptif.

Dans un second temps, ce chapitre s’intéresse au cas où le détecteur du chapitre 7 peut produire des faux positifs. Pour contrer cela, une gestion des faux positifs est proposée dans ce chapitre en répondant à l’objectif [E] du manuscrit. Le filtre particulaire discerne trois groupes de particules : le premier est celui qui valide la détection effectuée le recalage visuelle dessus ; le deuxième est celui qui ne valide pas cette même détection ; le troisième est composé des particules qui ont produit un mouvement illégal et doivent donc être régénérées. Cette gestion au système d’être moins dépendant des performances du détecteur tout en exploitant l’information provenant de celui-ci. Elle représente un compromis qui permet de surmonter une difficulté amenée par l’usage de réseau de neurones en tirant parti de l’information contenue dans la carte ainsi que des spécificités de l’information produite par les autres briques élémentaires du système. Les expériences menées montrent que le système peut gérer des situations présentant plus de difficultés que celles vues dans les chapitres précédents montrant une amélioration conséquente de la robustesse du système. La gestion des faux positifs par groupes est au centre de cette amélioration même quand le système est exposé à ses limites.

Chapitre 9

Bilan et conclusion généraux

Ce manuscrit présente une approche proposant de traiter la problématique de localisation en intérieur en exploitant la carte du bâtiment comme source d'information supplémentaire. Les travaux de cette thèse cherchent à construire un système dont le but est d'être embarqué dans le téléphone porté par le piéton afin de répondre aux objectifs [F & G]. Ce système est composé de briques élémentaires qui produisent une transformation ou une fusion de données pour obtenir une information plus raffinée. La conception de ce système répond à un cahier des charges résumé par les neuf objectifs mentionnés dans le chapitre 1 qui sont là pour solutionner les verrous scientifiques auxquels fait face cette problématique de localisation piétonne en intérieur.

Le manuscrit retrace l'acheminement de chaque brique depuis l'étape de conceptualisation jusqu'à l'intégration en passant par la validation. Toutes les briques élémentaires composant notre système ont été choisies en essayant de remplir les objectifs fixés. Elles ont été ensuite validées par une série de tests unitaires chacune avant d'être intégrées au système afin de mener les expériences de localisation. Les contributions de cette thèse se répartissent sur chacun de ces traitements.

La localisation en intérieur a pour particularité de s'effectuer dans un bâtiment ayant une carte. Cette carte est la source d'information sur laquelle ces travaux sont axés. C'est par l'extraction des contraintes géométriques et spatiales que la carte peut fournir l'information nécessaire pour assister à la localisation du piéton que ce soit en mode proprioceptif ou extéroceptif. Dans un premier temps, la carte répond directement à l'objectif de correction de l'estimation du cap en filtrant les particules du filtre particulaire commettant un mouvement ne respectant pas la physique du mouvement (i.e. traverser un mur). Ce filtrage par les murs de la carte a pour conséquence de corriger l'estimation de la position après un virage. Les objectifs [A & B] sont partiellement remplis grâce à l'introduction de ces contraintes de déplacement.

La représentation de la vraisemblance du mouvement humain sous la forme d'une grille est possible grâce à la discrétisation du plan du bâtiment. Les expériences menées avec cette grille de vraisemblance montrent qu'elle produit une contrainte de centrage sur le mouvement du piéton par rapport aux murs de la carte. Cette contrainte spatiale fait que les particules se dispersent moins dans l'espace fermé (i.e. un couloir). Leur propagation dans les espaces semi/ouverts est amoindrie du fait qu'initialement les particules étaient plus condensées quand elles ont quitté l'espace fermé. Ceci est montré en observant les ellipsoïdes d'incertitude aux différents endroits et en comparant l'usage ou non-usage de la grille de vraisemblance dans les deux cas. Cette représentation a pour effet de réduire l'incertitude de positionnement de

manière répétitive par la présence de murs. En sachant que la plupart des espaces intérieurs sont des arrangements de murs, cette contribution gagne en importance car elle remplit donc l'objectif [C]. D'autres parts, la grille de vraisemblance permet de renfoncer l'intégrité du système en garantissant à 99.7% que la position correcte du piéton se trouve dans l'ellipsoïde d'incertitude à trois écart-type ce qui resserre l'ellipsoïde.

Dans un second temps, la localisation proprioceptive connaît ses limites en l'absence d'éléments permettant de contraindre l'estimation du mouvement du piéton. C'est le cas où l'espace en intérieur présente une configuration ouverte importante (i.e. un hall ou une grande salle). Afin de toujours répondre aux objectifs [A, B & C], il est nécessaire d'apporter une information extéroceptive à partir d'un capteur disponible sur le téléphone. Dans notre étude, les solutions se basant sur l'infrastructure réseau du bâtiment ne sont pas en accord avec l'objectif [G] de proposer une solution embarquée et encapsulée. C'est pour cela que le recalage visuel par la caméra est plus attractif et qu'il a été choisi de le déployer.

Pour effectuer ce recalage visuel, la détection d'amers présents dans l'espace intérieur est nécessaire. Parmi les méthodes de détection d'objets à partir d'images, les réseaux de neurones montrent des performances remarquables. Une partie des travaux de cette thèse concernent la proposition d'une méthodologie d'adaptation d'un de ces modèles neuronaux à notre application. Cette méthodologie est en accord avec l'objectif [F] de déploiement sur téléphone et inclut les contraintes géométriques et spatiales tirées d'une carte.

La méthodologie d'adaptation répond au besoin de minimiser les coûts de construction d'un détecteur et d'entraînement sur des jeux de données. Cela nous a conduit à sélectionner parmi les modèles que propose l'état de l'art, un modèle neuronal de détection d'objets. Cette sélection s'accompagne d'un choix autour des amers à détecter. Une analyse de la distinguabilité, densité et standardisation par rapport à la carte du bâtiment est nécessaire pour définir les classes des amers à détecter. Quelque soit le modèle neuronal choisi, il est nécessaire de lui fournir un jeu données adapté au problème de détection traité. Cette méthodologie requiert la sélection ou la création du jeu données d'entraînement. Cette étape peut introduire un surcoût en temps et en argent d'où l'importance d'un choix judicieux des classes d'amers et de la stratégie d'entraînement. La stratégie d'entraînement du détecteur d'objet est conditionnée par le manque ou l'incomplétude du jeu de données. L'entraînement par transfert offre une réduction des coûts d'entraînement tout en gardant des résultats satisfaisants en entraînant le détecteur sur un jeu généraliste puis en le spécialisant sur le jeu de données dédié. À l'issue de cet entraînement, une campagne de tests pour valider est menée. Les détections produites après cette campagne peuvent contenir des cas problématiques. Il est nécessaire de les isoler et de proposer un traitement à ceux-ci. Notre méthodologie dicte qu'il faut décider pour chaque cas problématique s'il revient au détecteur de prendre en compte cette spécificité ou si c'est au processus en aval de gérer ces cas problématiques. Par exemple, la campagne de tests a montré que le détecteur avait une difficulté de différenciation en la classe cible : extincteur et une classe d'objets souvent présente dans l'espace intérieur : l'humain. La solution proposée est de permettre au modèle d'apprendre à détecter les deux classes. Le fait que le détecteur puisse détecter des humains a aussi pour but de répondre à un des objectifs de cette thèse qui reste en perspective. Cette méthodologie remplit totalement l'objectif [D].

La mise en œuvre du recalage visuel est proposée dans ce document de manière à inclure les contraintes spatiales de la carte du bâtiment dans chacune de ses étapes. Le recalage permet de garder les particules sur un cap correct lorsqu'une observation d'un amer est disponible. La mise à jour des particules vise à modifier le poids de chacune en fonction de l'écart à

l'angle dans lequel l'amer est observé. Cette instanciation du recalage montre une diminution conséquente de l'erreur de localisation et de l'incertitude cette erreur. Elle est aussi nécessaire dans le cas des expériences menées où le système dans sa forme proprioceptive échoue à la localisation du piéton tout le long de la trajectoire. Même si ce recalage visuel répond aux objectif [A, B & C], il n'est pas réaliste au sens où toutes les détections sont considérées comme étant des vrais positifs.

Dans la réalité, un réseau de neurones peut produire des faux positifs d'où la nécessité de répondre à l'objectif [E] qui concerne le renforcement de la localisation en intérieur par extraction des contraintes géométriques et spatiales de la carte du bâtiment pour l'élimination de faux positifs. Ces fausses alarmes ont différentes origines : elles peuvent être le résultat de similarités morphologique et caractéristique d'un amer cible avec un objet qui n'est pas considéré comme amer ; elles peuvent être visuellement correctes mais spatialement incohérentes comme dans le cas d'une réflexion ; ou elles peuvent aussi être inexplicables ce qui est une problématique propre au domaine de l'intelligence artificielle. Dans tous les cas, notre système a la capacité de gérer la possibilité que la détection qui lui est présentée soit fausse. C'est pour cela que la gestion de faux positifs est proposée par un artifice de validation des détections qui permet de diviser les particules en trois groupes dans le filtre particulaire. En exploitant les contraintes géométriques tirées de la carte, le processus de validation peut calculer la vraisemblance qu'accorde chaque particule à une détection afin que cette particule décide si elle croit ou non en la détection. Ce qui crée un groupe validant la détection et un autre qui ne valide par la détection. De plus, certaines particules ne respectant pas les contraintes du mouvement humain par rapport à la carte (i.e. franchissement de murs) sont marquées comme invalides et doivent être régénérées. Cette gestion des faux positifs lui permet de répondre à l'objectif [E] avec une limite du système liée à la disponibilité des amers dans l'espace intérieur.

Dans l'ensemble, le système proposé montre une précision de l'ordre de 25 cm dans des conditions favorables. Les amers dans cette configuration sont distribués de manière éparsée sur l'environnement intérieur. Leur présence est indispensable quand l'espace ouvert est de dimensions importantes. Les faux positifs présents parmi les détections sont de natures différentes et ont un effet différent sur la localisation. Certains de ces faux positifs sont aberrants et sont jugés peu vraisemblables par le système. Ils participent donc très peu voire aucunement au recalage visuel. Par conséquent, ils faussent l'information du système mais celle-ci reste à un niveau acceptable pour garder la localisation cohérente. Dans le cas où le système est exposé à des configurations plus difficiles à cause de l'agencement de l'environnement intérieur, du manque d'amers ou leur obstruction, et/ou la présence importante de faux positifs produits par YOLO, la précision est de l'ordre de 70 cm. Montrant que durant nos expériences, le système reste robuste même dans des cas défavorables jusqu'à atteindre ses limites et que la localisation devient imprécise. Parmi les neuf objectifs de départ, le déploiement sur téléphone [F], l'encapsulation du système et la réduction de sa consommation énergétique [G] et la localisation multi-agent par incorporation de comportement de déplacement [H] ne sont pas atteints pour des contraintes de temps et d'effort d'implantation. Cependant, ils ont servi de lignes directrices et justifient les choix pris concernant certaines briques élémentaires du systèmes. Et, ils aiguillent les perspectives de nos futurs travaux sur cette problématique.

De manière plus explicite, le choix des données inertielles comme source primaire de base pour la localisation est lié à l'objectif [F]. La disponibilité de cette information dans le téléphone est garantie car son capteur y est systématiquement embarqué. De plus le coût énergétique de ce capteur est faible au point d'être négligeable ce qui rejoint l'objectif [G]. La carte peut

être téléchargée dans le téléphone et la grille de vraisemblance peut être générée une seule fois pour être en lecture seule ce qui permet d'encapsuler le système. Le choix de la caméra en tant que source de signal extéroceptive est elle aussi lié à ces deux objectifs. L'une des questions qui se pose concerne l'étude de l'embarquabilité du modèle neuronal YOLO dans un téléphone. Si, celui là est déjà à la portée de la faisabilité et répond déjà au critère temps réel, son intégration dans notre système dans le cas où il est dans un téléphone portable reste à étudier. L'objectif [H] a guidé la définition de la représentation du mouvement humain ainsi que l'exigence de détection de la classe humain par le détecteur.

9.1 Perspectives

9.1.1 Embarquabilité et encapsulation sur téléphone portable

L'objectif [F & G] sont réalisables durant cette étape. L'embarquement de notre système requiert une réimplantation dans le langage de programmation du téléphone : Java/kotlin pour Android et Swift pour iOS. L'instanciation du filtre particulaire laisse à penser que le processus possède plusieurs optimisations logicielles possibles et un niveau de parallélisme assez important malgré certaines dépendances de données. Enfin, le modèle neuronal YOLO possède plusieurs versions et est paramétrable en taille. Ceci offre un compromis entre précision et vitesse d'inférence. De plus, il existe des moteurs d'inférence dont le but est d'optimiser cette étape sur du matériel dédié tel que NVidia TensorRT (GPU) ou OpenVINO d'Intel (CPU/FPGA). Ces moteurs annoncent une réduction significative du temps d'inférence et sont une piste à explorer.

9.1.2 Consommation énergétique

La localisation est un service qui, une fois mis à disposition sur un téléphone, partage les ressources systèmes et la réserve énergétique de l'appareil. Si ce service est énergivore, il présente un refus à l'adoption et l'utilisateur cherche alors à le désactiver. Donc, une étude sur la consommation énergétique est nécessaire. Si cette consommation pour l'acquisition des données IMUs et des images semble être négligeable, le traitement via le filtre particulaire et l'inférence par le réseau de neurones peuvent être énergivores. Un effort important à fournir pour estimer précisément la consommation de ce système dans un téléphone est requise afin de fournir à l'utilisateur un service adéquat.

9.1.3 Partage de carte dans un cadre multi-agent

Le système dans son état actuel peut localiser un piéton et le placer sur une carte. Si des exemplaires de ce système sont distribués sur plusieurs piétons, il est possible de les localiser indépendamment. Et ensuite, il est possible de les positionner sur la même carte. Le fait que le détecteur d'amers puisse détecter des humains peut permettre de les positionner par rapport au repère égocentré de la caméra. Une fusion des informations des localisations de différents piétons qui sont en vision les uns des autres est possible pour effectuer une localisation multi-agents collaborative. dans le cas où cette collaboration est centralisée, elle va à l'encontre de l'objectif [G] (encapsulation). Elle peut être localisée entre agents qui sont à proximité. Enfin, le fait que la grille de vraisemblance permette de décrire un comportement de marche

rend possible la description des comportements différents par piéton, ou des comportements contextuels à la situation, i.e. si deux piétons marchent côte à côte, ils peuvent influencer localement la vraisemblance de déplacement de l'autre. Il est aussi possible que la localisation collaborative intervienne pour la validation de détection des amers afin de gérer les faux positifs. Si plusieurs systèmes ont une détection, celle-ci est plus vraisemblable que si elle était seule. Cet axe d'étude présente plusieurs possibilités et paraît prometteur.

Bibliographie

- [Abadleh et al., 2017] Abadleh, A., Al-Hawari, E., Alkafaween, E., and Al-Sawalqah, H. (2017). Step detection algorithm for accurate distance estimation using dynamic step length. In *2017 18th IEEE international conference on mobile data management (MDM)*, pages 324–327. IEEE.
- [Ahwiadi and Wang, 2020] Ahwiadi, M. and Wang, W. (2020). An adaptive particle filter technique for system state estimation and prognosis. *IEEE Transactions on Instrumentation and Measurement*, 69(9) :6756–6765.
- [Akal et al., 2018] Akal, O., Mukherjee, T., Barbu, A., Paquet, J., George, K., and Pasiliao, E. (2018). A distributed sensing approach for single platform image-based localization. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 643–649. IEEE.
- [Alvarez et al., 2006] Alvarez, D., González, R. C., López, A., and Alvarez, J. C. (2006). Comparison of step length estimators from wearable accelerometer devices. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5964–5967. IEEE.
- [Antigny, 2018] Antigny, N. (2018). *Estimation continue de la pose d'un équipement tenu en main par fusion des données visio-inertielles pour les applications de navigation piétonne en milieux urbains*. PhD thesis, École centrale de Nantes.
- [Ariyanto and Nixon, 2011] Ariyanto, G. and Nixon, M. S. (2011). Model-based 3d gait biometrics. In *2011 international joint conference on biometrics (IJCB)*, pages 1–7. IEEE.
- [Armand and Tâche, 2022] Armand, P. and Tâche, J. (2022). 3d modelling and simulation of the dispersion of droplets and drops carrying the sars-cov-2 virus in a railway transport coach. *Scientific Reports*, 12(1) :1–22.
- [ARToolKit, 2004] ARToolKit (2004). <https://github.com/artoolkit>. consulté le 2023-05-09.
- [Árvai and Homolya, 2019] Árvai, L. and Homolya, S. (2019). Filtering and fingerprint matching methods for wi-fi radio map based indoor localization. In *2019 20th International Carpathian Control Conference (ICCC)*, pages 1–6. IEEE.
- [Ashraf et al., 2020] Ashraf, I., Hur, S., and Park, Y. (2020). Smartphone sensor based indoor positioning : Current status, opportunities, and future challenges. *Electronics*, 9(6) :891.
- [Awarkeh et al., 2018] Awarkeh, N., Cousin, J.-C., Muller, M., and Samama, N. (2018). Improvement of the angle measurement accuracy for indoor uwb localization. In *2018 15th European Radar Conference (EuRAD)*, pages 285–288. IEEE.
- [Azam et al., 2022] Azam, M. A., Sampieri, C., Ioppi, A., Africano, S., Vallin, A., Mocellin, D., Fragale, M., Guastini, L., Moccia, S., Piazza, C., et al. (2022). Deep learning applied

- to white light and narrow band imaging videolaryngoscopy : Toward real-time laryngeal cancer detection. *The Laryngoscope*, 132(9) :1798–1806.
- [Baek et al., 2019] Baek, F., Ha, I., and Kim, H. (2019). Augmented reality system for facility management using image-based indoor localization. *Automation in construction*, 99 :18–26.
- [Bai et al., 2014] Bai, Y., Jia, W., Zhang, H., Mao, Z.-H., and Sun, M. (2014). Landmark-based indoor positioning for visually impaired individuals. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 668–671. IEEE.
- [Baker et al., 2017] Baker, R., Leboeuf, F., Reay, J., and Sangeux, M. (2017). *The Conventional Gait Model : The Success and Limitations*, pages 1–19. Springer International.
- [Barsocchi et al., 2013] Barsocchi, P., Chessa, S., Furfari, F., and Potortì, F. (2013). Evaluating aal solutions through competitive benchmarking : the localization competition. *IEEE Pervasive Computing Magazine*, 12(4) :72–79.
- [Barsocchi et al., 2012] Barsocchi, P., Potortì, F., Furfari, F., and Gil, A. M. M. (2012). Comparing aal indoor localization systems. In *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking : International Competition, EvAAL 2011, Competition in Valencia, Spain, July 25-29, 2011, and Final Workshop in Lecce, Italy, September 26, 2011. Revised Selected Papers 1*, pages 1–13. Springer.
- [Bassett et al., 2017] Bassett, D. R., Toth, L. P., LaMunion, S. R., and Crouter, S. E. (2017). Step counting : a review of measurement considerations and health-related applications. *Sports Medicine*, 47 :1303–1315.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. *Lecture notes in computer science*, 3951 :404–417.
- [Beauregard et al., 2008] Beauregard, S., Klepal, M., et al. (2008). Indoor pdr performance enhancement using minimal map information and particle filters. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 141–147. IEEE.
- [Benligiray et al., 2019] Benligiray, B., Topal, C., and Akinlar, C. (2019). Stag : A stable fiducial marker system. *Image and Vision Computing*, 89 :158–169.
- [Bentley - Legion Simulator, 2023] Bentley - Legion Simulator (2023). <https://www.bentley.com/software/legion-simulator/>. consulté le 2023-05-01.
- [Bergamasco et al., 2016] Bergamasco, F., Albarelli, A., Cosmo, L., Rodolà, E., and Torsello, A. (2016). An accurate and robust artificial marker based on cyclic codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12) :2359–2373.
- [Binelli et al., 2023] Binelli, M. R., van Dommelen, R., Nagel, Y., Kim, J., Haque, R. I., Coulter, F. B., Siqueira, G., Studart, A. R., and Briand, D. (2023). Digital manufacturing of personalised footwear with embedded sensors. *Scientific Reports*, 13(1) :1962.
- [Blok et al., 2019] Blok, P. M., van Boheemen, K., van Evert, F. K., IJsselmuiden, J., and Kim, G.-H. (2019). Robot navigation in orchards with localization based on particle filter and kalman filter. *Computers and electronics in agriculture*, 157 :261–269.
- [Bo Bo et al., 2015] Bo Bo, N., Deboeverie, F., Veelaert, P., and Philips, W. (2015). Real-time multi-people tracking by greedy likelihood maximization. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, pages 32–37.
- [Bochkovskiy et al., 2020] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*.

- [Bolic et al., 2005] Bolic, M., Djuric, P. M., and Sangjin Hong (2005). Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, 53(7) :2442–2450.
- [Brajdic and Harle, 2013] Brajdic, A. and Harle, R. (2013). Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 225–234.
- [Brossard et al., 2020a] Brossard, M., Barrau, A., and Bonnabel, S. (2020a). Ai-imu dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4) :585–595.
- [Brossard et al., 2020b] Brossard, M., Bonnabel, S., and Barrau, A. (2020b). Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3) :4796–4803.
- [Burkart and Huber, 2021] Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70 :245–317.
- [Campos et al., 2020] Campos, C., Elvira, R., Gómez, J. J., Montiel, J. M. M., and Tardós, J. D. (2020). ORB-SLAM3 : An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv preprint arXiv :2007.11898*.
- [Carpenter et al., 1999] Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1) :2–7.
- [Caso et al., 2018] Caso, G., Le, M., De Nardis, L., and Di Benedetto, M.-G. (2018). Performance comparison of wifi and uwb fingerprinting indoor positioning systems. *Technologies*, 6(1) :14.
- [Cavagna et al., 1976] Cavagna, G. A., Thys, H., and Zamboni, A. (1976). The sources of external work in level walking and running. *The Journal of physiology*, 262(3) :639–657.
- [Chen et al., 2018] Chen, C., Zhao, P., Lu, C. X., Wang, W., Markham, A., and Trigoni, N. (2018). Oxiod : The dataset for deep inertial odometry. <http://deepio.cs.ox.ac.uk/>.
- [Chen and Clarke, 2020] Chen, J. and Clarke, K. C. (2020). Indoor cartography. *Cartography and Geographic Information Science*, 47(2) :95–109.
- [Cheraghi et al., 2021] Cheraghi, S. A., Fusco, G., and Coughlan, J. M. (2021). Real-time sign detection for accessible indoor navigation. In *Journal on technology and persons with disabilities :... Annual International Technology and Persons with Disabilities Conference*, volume 9, page 125. NIH Public Access.
- [Cheraghi et al., 2019] Cheraghi, S. A., Sharma, A., Namboodiri, V., and Arsal, G. (2019). Safeexit4aai : An inclusive indoor emergency evacuation system for people with disabilities. In *Proceedings of the 16th International Web for All Conference, W4A '19*, New York, NY, USA. Association for Computing Machinery.
- [Chghaf et al., 2022] Chghaf, M., Rodriguez, S., and Ouardi, A. E. (2022). Camera, lidar and multi-modal slam systems for autonomous ground vehicles : a survey. *Journal of Intelligent & Robotic Systems*, 105(1) :2.
- [Chollet and Liu, 2013] Chollet, F. and Liu, H. (2013). A (not so) short introduction to micro electro mechanical systems. *MEMS Encyclopedia*.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20 :273–297.

- [Dang et al., 2015] Dang, V. C., Kubo, M., Sato, H., Shirakawa, T., and Namatame, A. (2015). Occupancy grid map of semi-static objects by mobile observer. *Artificial Life and Robotics*, 20(1) :7–12.
- [Das and Mao, 2020] Das, S. and Mao, E. (2020). The global energy footprint of information and communication technology electronics in connected internet-of-things devices. *Sustainable Energy, Grids and Networks*, 24 :100408.
- [de Souza Cardoso et al., 2020] de Souza Cardoso, L. F., Mariano, F. C. M. Q., and Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers & Industrial Engineering*, 139 :106159.
- [Dehghan Firoozabadi et al., 2019] Dehghan Firoozabadi, A., Azurdia-Meza, C., Soto, I., Seguel, F., Krommenacker, N., Iturralde, D., Charpentier, P., and Zabala-Blanco, D. (2019). A novel frequency domain visible light communication (vlc) three-dimensional trilateration system for localization in underground mining. *Applied Sciences*, 9(7) :1488.
- [Deng et al., 2017] Deng, Z., Si, W., Qu, Z., Liu, X., and Na, Z. (2017). Heading estimation fusing inertial sensors and landmarks for indoor navigation using a smartphone in the pocket. *EURASIP Journal on Wireless Communications and Networking*, 2017(1) :160.
- [Deng et al., 2015] Deng, Z.-A., Wang, G., Hu, Y., and Wu, D. (2015). Heading estimation for indoor pedestrian navigation using a smartphone in the pocket. *Sensors (Basel, Switzerland)*, 15 :21518–21536.
- [Desai and Rattan, 2009] Desai, P. and Rattan, K. S. (2009). Indoor localization and surveillance using wireless sensor network and pan/tilt camera. In *Proceedings of the IEEE 2009 National Aerospace & Electronics Conference (NAECON)*, pages 1–6. IEEE.
- [Díez et al., 2018] Díez, L. E., Bahillo, A., Otegui, J., and Otim, T. (2018). Step length estimation methods based on inertial sensors : A review. *IEEE Sensors Journal*, 18(17) :6908–6926.
- [Dobre et al., 2017] Dobre, C., Mavromoustakis, C. X., Garcia, N. M., Mastorakis, G., and Goleva, R. I. (2017). Introduction to the aal and ele systems. In *Ambient assisted living and enhanced living environments*, pages 1–16. Elsevier.
- [Dong et al., 2021] Dong, L., Xu, J., Chi, G., Li, D., Zhang, X., Li, J., Ma, Q., and Yang, Z. (2021). Enabling surveillance cameras to navigate. *ACM Transactions on Sensor Networks (TOSN)*, 17(4) :1–20.
- [Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*.
- [Douc and Cappe, 2005] Douc, R. and Cappe, O. (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.
- [Elfring et al., 2021] Elfring, J., Torta, E., and van de Molengraft, R. (2021). Particle filters : A hands-on tutorial. *Sensors*, 21(2).
- [Elvira et al., 2017] Elvira, V., Míguez, J., and Djurić, P. M. (2017). Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment. *IEEE Transactions on Signal Processing*, 65(7) :1781–1794.

- [Endres et al., 2013] Endres, F., Hess, J., Sturm, J., Cremers, D., and Burgard, W. (2013). 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, 30(1) :177–187.
- [Euston et al., 2008] Euston, M., Coote, P., Mahony, R., Kim, J., and Hamel, T. (2008). A complementary filter for attitude estimation of a fixed-wing uav. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 340–345. IEEE.
- [Fan et al., 2019] Fan, Q., Zhang, H., Pan, P., Zhuang, X., Jia, J., Zhang, P., Zhao, Z., Zhu, G., and Tang, Y. (2019). Improved pedestrian dead reckoning based on a robust adaptive kalman filter for indoor inertial location system. *Sensors*, 19(2) :294.
- [Faragher and Harle, 2015] Faragher, R. and Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE Journal on Selected Areas in Communications*, 33(11) :2418–2428.
- [Felzenszwalb and Huttenlocher, 2012] Felzenszwalb, P. F. and Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of computing*, 8(1) :415–428.
- [Frosio et al., 2008] Frosio, I., Pedersini, F., and Borghese, N. A. (2008). Autocalibration of mems accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 58(6) :2034–2041.
- [Fusco et al., 2020] Fusco, G., Cheraghi, S. A., Neat, L., and Coughlan, J. M. (2020). An indoor navigation app using computer vision and sign recognition. In *International Conference on Computers Helping People with Special Needs*, pages 485–494. Springer.
- [Gang and Pyun, 2019] Gang, H.-S. and Pyun, J.-Y. (2019). A smartphone indoor positioning system using hybrid localization technology. *Energies*, 12(19) :3702.
- [Garrido-Jurado et al., 2014] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6) :2280–2292.
- [Geiger et al., 2013] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics : The kitti dataset. *The International Journal of Robotics Research*, 32(11) :1231–1237.
- [Gelb et al., 1974] Gelb, A. et al. (1974). *Applied optimal estimation*. MIT press.
- [Gerstweiler, 2018] Gerstweiler, G. (2018). Guiding people in complex indoor environments using augmented reality. In *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*, pages 801–802. IEEE.
- [Ghaoui et al., 2022] Ghaoui, M. A., Vincke, B., and Reynaud, R. (2022). Human motion likelihood representation map-aided pdr particle filter. *IEEE Sensors Journal*.
- [Ghaoui et al., 2023] Ghaoui, M. A., Vincke, B., and Reynaud, R. (2023). Renforcement de la localisation piétonne par détection visuelle via des informations géométriques et spatiales contenues dans une carte. In *GRETSI 2023*, Grenoble, France. colloque GRETSI.
- [Gholamhosseini et al., 2019] Gholamhosseini, L., Sadoughi, F., and Safaei, A. (2019). Hospital real-time location system (a practical approach in healthcare) : A narrative review article. *Iranian journal of public health*, 48(4) :593.
- [Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

- [Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- [Gordon et al., 1993] Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140-2, pages 107–113. IET.
- [Gu et al., 2019] Gu, F., Hu, X., Ramezani, M., Acharya, D., Khoshelham, K., Valaee, S., and Shang, J. (2019). Indoor localization improved by spatial context—a survey. *ACM Comput. Surv.*, 52(3).
- [Gu et al., 2019] Gu, F., Khoshelham, K., Yu, C., and Shang, J. (2019). Accurate step length estimation for pedestrian dead reckoning localization using stacked autoencoders. *IEEE Transactions on Instrumentation and Measurement*, 68(8) :2705–2713.
- [Gu et al., 2015] Gu, Y., Song, Q., Li, Y., and Ma, M. (2015). Foot-mounted pedestrian navigation based on particle filter with an adaptive weight updating strategy. *Journal of Navigation*, 68(1) :23–38.
- [Guerra et al., 2019] Guerra, W., Tal, E., Murali, V., Ryou, G., and Karaman, S. (2019). Flightgoggles : Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6941–6948. IEEE.
- [Hafner et al., 2013] Hafner, P., Moder, T., Wieser, M., and Bernoulli, T. (2013). Evaluation of smartphone-based indoor positioning using different bayes filters. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10. IEEE.
- [Hayward et al., 2021] Hayward, S., Earps, J., Sharpe, R., van Lopik, K., Tribe, J., and West, A. (2021). A novel inertial positioning update method, using passive rfid tags, for indoor asset localisation. *CIRP Journal of Manufacturing Science and Technology*, 35 :968–982.
- [He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- [Hoag, 1963] Hoag, D. (1963). Apollo guidance and navigation : Considerations of apollo imu gimbal lock. *Cambridge : MIT Instrumentation Laboratory*, pages 1–64.
- [Hol et al., 2006] Hol, J. D., Schon, T. B., and Gustafsson, F. (2006). On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82.
- [Hou and Bergmann, 2020a] Hou, X. and Bergmann, J. (2020a). A pedestrian dead reckoning method for head-mounted sensors. *Sensors*, 20(21) :6349.
- [Hou and Bergmann, 2020b] Hou, X. and Bergmann, J. (2020b). Pedestrian dead reckoning with wearable sensors : A systematic review. *IEEE Sensors Journal*, 21(1) :143–152.
- [Hu et al., 2020] Hu, G., Zhang, W., Wan, H., and Li, X. (2020). Improving the heading accuracy in indoor pedestrian navigation based on a decision tree and kalman filter. *Sensors*, 20(6) :1578.
- [Hu et al., 2017] Hu, Z., Huang, G., Hu, Y., and Yang, Z. (2017). Wi-vi fingerprint : Wifi and vision integrated fingerprint for smartphone-based indoor self-localization. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4402–4406. IEEE.

- [Huai et al., 2019] Huai, J., Zhang, Y., and Yilmaz, A. (2019). The mobile ar sensor logger for android and ios devices. In *2019 IEEE SENSORS*.
- [Huang et al., 2011] Huang, J., Millman, D., Quigley, M., Stavens, D., Thrun, S., and Aggarwal, A. (2011). Efficient, generalized indoor wifi graphslam. In *2011 IEEE international conference on robotics and automation*, pages 1038–1043. IEEE.
- [Huang et al., 2020] Huang, L., Yu, B., Li, H., Zhang, H., Li, S., Zhu, R., and Li, Y. (2020). Hpips : A high-precision indoor pedestrian positioning system fusing wifi-rtt, mems, and map information. *Sensors*, 20(23) :6795.
- [Huang et al., 2018] Huang, Z., Gu, N., Hao, J., and Shen, J. (2018). 3dloc : 3d features for accurate indoor positioning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4) :1–26.
- [Jadon et al., 2019] Jadon, A., Omama, M., Varshney, A., Ansari, M. S., and Sharma, R. (2019). Firenet : a specialized lightweight fire & smoke detection model for real-time iot applications. *arXiv preprint arXiv :1905.11922*.
- [Jardosh et al., 2009] Jardosh, A. P., Papagiannaki, K., Belding, E. M., Almeroth, K. C., Iannaccone, G., and Vinnakota, B. (2009). Green wlans : on-demand wlan infrastructures. *Mobile Networks and Applications*, 14 :798–814.
- [Jayalath et al., 2013] Jayalath, S., Abhayasinghe, N., and Murray, I. (2013). A gyroscope based accurate pedometer algorithm. In *International Conference on Indoor Positioning and Indoor Navigation*, volume 28, page 31st.
- [Jiang et al., 2022] Jiang, P., Ergu, D., Liu, F., Cai, Y., and Ma, B. (2022). A review of yolo algorithm developments. *Procedia Computer Science*, 199 :1066–1073.
- [Jimenez et al., 2009] Jimenez, A. R., Seco, F., Prieto, C., and Guevara, J. (2009). A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42.
- [Jocher et al., 2020] Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R. (2020). yolov5. *Code repository www.github.com/ultralytics/yolov5*.
- [Joshi et al., 2021] Joshi, G., Walambe, R., and Kotecha, K. (2021). A review on explainability in multimodal deep neural nets. *IEEE Access*, 9 :59800–59821.
- [Jouin et al., 2016] Jouin, M., Gouriveau, R., Hissel, D., Péra, M.-C., and Zerhouni, N. (2016). Particle filter-based prognostics : Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72-73 :2–31.
- [Ju et al., 2018] Ju, H., Park, S. Y., and Park, C. G. (2018). A smartphone-based pedestrian dead reckoning system with multiple virtual tracking for indoor navigation. *IEEE Sensors Journal*, 18(16) :6756–6764.
- [Julier and Uhlmann, 1997] Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. Spie.
- [Kainz et al., 2017] Kainz, H., Graham, D., Edwards, J., Walsh, H. P., Maine, S., Boyd, R. N., Lloyd, D. G., Modenese, L., and Carty, C. P. (2017). Reliability of four models for clinical gait analysis. *Gait & posture*, 54 :325–331.
- [Kalman et al., 1960] Kalman, R. E. et al. (1960). Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2) :102–119.

- [Kamiya et al., 2019] Kamiya, Y., Gu, Y., and Kamijo, S. (2019). Indoor positioning in large shopping mall with context based map matching. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6.
- [Karkus et al., 2018] Karkus, P., Hsu, D., and Lee, W. S. (2018). Particle filter networks with application to visual localization. In *Conference on robot learning*, pages 169–178. PMLR.
- [Ke et al., 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm : A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- [Keller et al., 1985] Keller, J. M., Gray, M. R., and Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4) :580–585.
- [Khedr and El-Sheimy, 2017] Khedr, M. and El-Sheimy, N. (2017). A smartphone step counter using imu and magnetometer for navigation and health monitoring applications. *Sensors*, 17(11) :2573.
- [Kim et al., 2020] Kim, J.-a., Sung, J.-Y., and Park, S.-h. (2020). Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. In *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)*, pages 1–4. IEEE.
- [Kim et al., 2004] Kim, J. W., Jang, H. J., Hwang, D.-H., and Park, C. (2004). A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3(1-2) :273–279.
- [Klein and Asraf, 2020] Klein, I. and Asraf, O. (2020). Stepnet—deep learning approaches for step length estimation. *IEEE Access*, 8 :85706–85713.
- [Klein et al., 2018] Klein, I., Solaz, Y., and Ohayon, G. (2018). Pedestrian dead reckoning with smartphone mode recognition. *IEEE Sensors Journal*, 18(18) :7577–7584.
- [Kleinmeier et al., 2019] Kleinmeier, B., Zönnchen, B., Gödel, M., and Köster, G. (2019). VADERE : An open-source simulation framework to promote interdisciplinary understanding. *arXiv preprint arXiv :1907.09520*.
- [Klepal et al., 2008] Klepal, M., Beauregard, S., et al. (2008). A backtracking particle filter for fusing building plans with pdr displacement estimates. In *2008 5th Workshop on Positioning, Navigation and Communication*, pages 207–212. IEEE.
- [Kong et al., 1994] Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425) :278–288.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [Kunhoth et al., 2020] Kunhoth, J., Karkar, A., Al-Maadeed, S., and Al-Ali, A. (2020). Indoor positioning and wayfinding systems : a survey. *Human-centric Computing and Information Sciences*, 10(1) :1–41.
- [La Rousse, 2023] La Rousse (2023). Dictionnaire. <https://www.larousse.fr/dictionnaires/francais/pas/58420>. consulté le 2023-05-01.
- [Leboeuf et al., 2019] Leboeuf, F., Baker, R., Barré, A., Reay, J., Jones, R., and Sangeux, M. (2019). The conventional gait model, an open-source implementation that reproduces the past but prepares for the future. *Gait & posture*, 69 :235–241.

- [Li et al., 2022a] Li, C., Chai, W., and Li, Q. (2022a). Multi-sensor fusion-based indoor single-track semantic map construction and localization.
- [Li et al., 2022b] Li, C., Chai, W., Yang, X., and Li, Q. (2022b). Crowdsourcing-based indoor semantic map construction and localization using graph optimization. *Sensors*, 22(16).
- [Li et al., 2018] Li, J., Bi, Y., Li, K., Wang, K., Lin, F., and Chen, B. M. (2018). Accurate 3d localization for mav swarms by uwb and imu fusion. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pages 100–105. IEEE.
- [Li et al., 2020] Li, M., Zhang, Z., Lei, L., Wang, X., and Guo, X. (2020). Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks : Comparison of faster r-cnn, yolo v3 and ssd. *Sensors*, 20(17) :4938.
- [Liang et al., 2018] Liang, Q., Wang, L., Li, Y., and Liu, M. (2018). Indoor mapping and localization for pedestrians using opportunistic sensing with smartphones. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1649–1656.
- [Liaw et al., 2002] Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3) :18–22.
- [Lin et al., 2020] Lin, J., Zou, C., Lan, L., Gu, S., and An, X. (2020). Deep heading estimation for pedestrian dead reckoning. *Journal of Physics : Conference Series*, 1656 :012009.
- [Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco : Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- [Liu et al., 2021] Liu, Y., Sun, P., Wergeles, N., and Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172 :114602.
- [Lorincz et al., 2019] Lorincz, J., Capone, A., and Wu, J. (2019). Greener, energy-efficient and sustainable networks : State-of-the-art and new trends.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60 :91–110.
- [Lu et al., 2019] Lu, C., Uchiyama, H., Thomas, D., Shimada, A., and Taniguchi, R.-i. (2019). Indoor positioning system based on chest-mounted imu. *Sensors*, 19(2) :420.
- [Lu et al., 2017] Lu, G., Yan, Y., Sebe, N., and Kambhamettu, C. (2017). Indoor localization via multi-view images and videos. *Computer Vision and Image Understanding*, 161 :145–160.
- [Lu, 2018] Lu, X. X. (2018). A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics : Conference Series*, volume 1087, page 052009. IOP Publishing.
- [Luo et al., 2020a] Luo, J., Zhang, C., and Wang, C. (2020a). Indoor multi-floor 3d target tracking based on the multi-sensor fusion. *IEEE Access*, 8 :36836–36846.
- [Luo et al., 2020b] Luo, Y., Guo, C., Su, J., Guo, W., and Zhang, Q. (2020b). Learning based complex motion patterns recognition for pedestrian dead reckoning. *IEEE Sensors Journal*, pages 1–1.
- [Luperto et al., 2020] Luperto, M., Antonazzi, M., Amigoni, F., and Borghese, N. A. (2020). Robot exploration of indoor environments using incomplete and inaccurate prior knowledge. *Robotics and Autonomous Systems*, 133 :103622.

- [Macario Barros et al., 2022] Macario Barros, A., Michel, M., Moline, Y., Corre, G., and Carrel, F. (2022). A comprehensive survey of visual slam algorithms. *Robotics*, 11(1) :24.
- [Madgwick et al., 2011] Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, R. (2011). Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7.
- [Mahony et al., 2008] Mahony, R., Hamel, T., and Pflimlin, J.-M. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5) :1203–1218.
- [Marchand et al., 2015] Marchand, E., Uchiyama, H., and Spindler, F. (2015). Pose estimation for augmented reality : a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12) :2633–2651.
- [Marouane et al., 2014] Marouane, C., Maier, M., Feld, S., and Werner, M. (2014). Visual positioning systems—an extension to movips. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 95–104. IEEE.
- [Mass motion, 2023] Mass motion (2023). <https://www.arup.com/services/digital/massmotion>. consulté le 2023-05-01.
- [Matt, 2017a] Matt, M. (2017a). How is arcore better than arkit? <https://medium.com/6d-ai/how-is-arcore-better-than-arkit-5223e6b3e79d>. consulté le 2023-05-01.
- [Matt, 2017b] Matt, M. (2017b). Why is arkit better than the alternatives? <https://medium.com/6d-ai/why-is-arkit-better-than-the-alternatives-af8871889d6a>. consulté le 2023-05-01.
- [McNamara, 2021] McNamara, D. (2021). How have phone sizes changed over the years? <https://www.thestreet.com/review/how-have-phone-sizes-changed-over-the-years/>. consulté le 2023-05-12.
- [Mekhalfi et al., 2021] Mekhalfi, M. L., Nicolò, C., Bazi, Y., Al Rahhal, M. M., Alsharif, N. A., and Al Maghayreh, E. (2021). Contrasting yolov5, transformer, and efficientdet detectors for crop circle detection in desert. *IEEE Geoscience and Remote Sensing Letters*, 19 :1–5.
- [Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6) :1087–1092.
- [Meyer-Delius et al., 2012] Meyer-Delius, D., Beinhofer, M., and Burgard, W. (2012). Occupancy grid models for robot mapping in changing environments. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- [Michel et al., 2015] Michel, T., Fourati, H., Geneves, P., and Layaïda, N. (2015). A comparative analysis of attitude estimation for pedestrian navigation with smartphones. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10. IEEE.
- [Michel et al., 2017] Michel, T., Geneves, P., Fourati, H., and Layaïda, N. (2017). On attitude estimation with smartphones. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 267–275. IEEE.
- [Michel et al., 2018] Michel, T., Genevès, P., Fourati, H., and Layaïda, N. (2018). Attitude Estimation for Indoor Navigation and Augmented Reality with Smartphones. *Pervasive and Mobile Computing*, 46 :96–121.

- [Mićović et al., 2021] Mićović, A., Mladan, D., and Brkljač, N. (2021). Fire extinguisher conformity assessment—a case study. In *Thematic conference proceedings of international significance [Elektronski izvor]/International Scientific Conference "Archibald Reiss Days", Belgrade, 9-10 November 2021*, pages 575–588. Belgrade : University of Criminal Investigation and Police Studies.
- [Mikolajczyk, 2021] Mikolajczyk, A. (2021). AgaMiko/data-augmentation-review : A comprehensive review of data augmentation resources.
- [Mingyuan et al., 2019] Mingyuan, M., Hewei, Z., Simeng, L., and Baochang, Z. (2019). Semantic-rtab-map (srm) : A semantic slam system with cnns on depth images. *Mathematical Foundations of Computing*, 2(A0000-0001_2019_1_29) :29.
- [Morar et al., 2020] Morar, A., Moldoveanu, A., Mocanu, I., Moldoveanu, F., Radoi, I. E., Asavei, V., Gradinaru, A., and Butean, A. (2020). A comprehensive survey of indoor localization methods based on computer vision. *Sensors*, 20(9) :2641.
- [Motroni et al., 2018] Motroni, A., Nepa, P., Magnago, V., Buffi, A., Tellini, B., Fontanelli, D., and Macii, D. (2018). Sar-based indoor localization of uhf-rfid tags via mobile robot. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8.
- [Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orbslam : a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5) :1147–1163.
- [Mur-Artal and Tardós, 2017] Mur-Artal, R. and Tardós, J. D. (2017). Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2) :796–803.
- [Murata et al., 2019] Murata, M., Ahmetovic, D., Sato, D., Takagi, H., Kitani, K. M., and Asakawa, C. (2019). Smartphone-based localization for blind navigation in building-scale indoor environments. *Pervasive and Mobile Computing*, 57 :14 – 32.
- [Murray, 2012] Murray, L. (2012). Gpu acceleration of the particle filter : the metropolis resampler. *arXiv preprint arXiv :1202.6163*.
- [Murray et al., 2016] Murray, L. M., Lee, A., and Jacob, P. E. (2016). Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3) :789–805.
- [Narejo et al., 2021] Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C., and Anjum, M. R. (2021). Weapon detection using yolo v3 for smart surveillance system. *Mathematical Problems in Engineering*, 2021 :1–9.
- [Neethirajan, 2022] Neethirajan, S. (2022). Chicktrack – a quantitative tracking tool for measuring chicken activity. *Measurement*, 191 :110819.
- [Nepal and Eslamiat, 2022] Nepal, U. and Eslamiat, H. (2022). Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. *Sensors*, 22(2).
- [Nicholas, 2022] Nicholas, R. (2022). Pose method. <https://posemethod.com/usain-bolts-running-technique/>.
- [Nicholson et al., 2018] Nicholson, L., Milford, M., and Sünderhauf, N. (2018). Quadricslam : Constrained dual quadrics from object detections as landmarks in semantic slam. *IEEE Robotics and Automation Letters (RA-L)*.
- [Ning and Chen, 2019] Ning, F.-S. and Chen, Y.-C. (2019). Combining a modified particle filter method and indoor magnetic fingerprint map to assist pedestrian dead reckoning for indoor positioning and navigation. *Sensors*, 20(1) :185.

- [Oasys, 2023] Oasys (2023). <https://www.oasys-software.com/solutions/pedestrian-simulation/>. consulté le 2023-05-01.
- [Obeidat et al., 2021] Obeidat, H., Shuaieb, W., Obeidat, O., and Abd-Alhameed, R. (2021). A review of indoor localization techniques and wireless technologies. *Wireless Personal Communications*, 119 :289–327.
- [Ooi et al., 2018] Ooi, B.-Y., Lee, W.-K., and Chea, K.-C. (2018). Localization of mobile sensor nodes using qr codes and dead reckoning with error correction. In *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, pages 841–845. IEEE.
- [OpenAI, 2023] OpenAI (2023). Gpt-4 technical report. *arXiv*.
- [Orendurff et al., 2008] Orendurff, M. S., Schoen, J. A., Bernatz, G. C., Segal, A. D., and Klute, G. K. (2008). How humans walk : bout duration, steps per bout, and rest duration. *Journal of Rehabilitation Research & Development*, 45(7).
- [Oudre et al., 2018] Oudre, L., Barrois-Müller, R., Moreau, T., Truong, C., Vienne-Jumeau, A., Ricard, D., Vayatis, N., and Vidal, P.-P. (2018). Template-based step detection with inertial measurement units. *Sensors*, 18(11) :4033.
- [Pan and Tompkins, 1985] Pan, J. and Tompkins, W. J. (1985). A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3) :230–236.
- [Pham et al., 2003] Pham, D., Dahia, K., and Musso, C. (2003). A kalman-particle kernel filter and its application to terrain navigation. *Proceedings 6th ICIF*, page 24.
- [Placed et al., 2023] Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., and Castellanos, J. A. (2023). A survey on active simultaneous localization and mapping : State of the art and new frontiers. *IEEE Transactions on Robotics*.
- [Potortì et al., 2015] Potortì, F., Barsocchi, P., Girolami, M., Torres-Sospedra, J., and Montoliu, R. (2015). Evaluating indoor localization solutions in large environments through competitive benchmarking : The evaal-etri competition. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10. IEEE.
- [Potortì et al., 2017] Potortì, F., Park, S., Jimenez Ruiz, A. R., Barsocchi, P., Girolami, M., Crivello, A., Lee, S. Y., Lim, J. H., Torres-Sospedra, J., Seco, F., et al. (2017). Comparing the performance of indoor localization systems through the evaal framework. *Sensors*, 17(10) :2327.
- [Potortì, 2020] Potortì, F. (2020). The ipin 2019 indoor localisation competition—description and results. *IEEE Access*, 8 :206674–206718.
- [Priyantha, 2005] Priyantha, N. B. (2005). *The cricket indoor location system*. PhD thesis, Massachusetts Institute of Technology.
- [PTV Viswalk, 2023] PTV Viswalk (2023). <https://www.myptv.com/en/mobility-software/pedestrian-simulation-software-ptv-viswalk>. consulté le 2023-05-01.
- [Qiu et al., 2021] Qiu, J., Lo, F. P.-W., Gu, X., Sun, Y., Jiang, S., and Lo, B. (2021). Indoor future person localization from an egocentric wearable camera. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8586–8592. IEEE.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

- [Redmon and Farhadi, 2017] Redmon, J. and Farhadi, A. (2017). Yolo9000 : better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.
- [Redmon and Farhadi, 2018] Redmon, J. and Farhadi, A. (2018). Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- [Renaudin, 2018] Renaudin, V. (2018). Uliss and persy : Research equipment for the positioning and navigation of travelers.
- [Renaudin, 2019] Renaudin, V. (2019). Evaluating indoor positioning systems in a shopping mall : The lessons learned from the ipin 2018 competition. *IEEE Access*, 7 :148594–148628.
- [Renaudin et al., 2010] Renaudin, V., Afzal, M. H., and Lachapelle, G. (2010). New method for magnetometers based orientation estimation. In *IEEE/ION Position, Location and Navigation Symposium*, pages 348–356. IEEE.
- [Richards, 2020] Richards, T. (2020). A review of software for crowd simulation. *Leeds Institute for Data Science, Tech. Rep.[Online]*. Available : <https://urbananalytics.github.io/dust/docs/pedsimreview.pdf>.
- [Rish et al., 2001] Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.
- [Rosenfeld, 2012] Rosenfeld, B. A. (2012). *A history of non-Euclidean geometry : Evolution of the concept of a geometric space*, volume 12. Springer Science & Business Media.
- [Rosinol et al., 2020] Rosinol, A., Abate, M., Chang, Y., and Carlone, L. (2020). Kimera : an open-source library for real-time metric-semantic localization and mapping.
- [Roy and Chowdhury, 2022] Roy, P. and Chowdhury, C. (2022). A survey on ubiquitous wifi-based indoor localization system for smartphone users from implementation perspectives. *CCF Transactions on Pervasive Computing and Interaction*, 4(3) :298–318.
- [Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb : An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee.
- [Saarinen et al., 2012] Saarinen, J., Andreasson, H., and Lilienthal, A. J. (2012). Independent markov chain occupancy grid maps for representation of dynamic environment. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3489–3495.
- [Sabatini, 2011] Sabatini, A. M. (2011). Kalman-filter-based orientation determination using inertial/magnetic sensors : Observability analysis and performance evaluation. *Sensors*, 11(10) :9182–9206.
- [Scarlett, 2007] Scarlett, J. (2007). Enhancing the performance of pedometers using a single accelerometer. *Application Note, Analog Devices*, 41.
- [Schenkluhn et al., 2023] Schenkluhn, M., Peukert, C., and Weinhardt, C. (2023). Augmented reality-based indoor positioning for smart home automations. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–6.

- [Schneider, 2018] Schneider, C. J. (2018). Body worn cameras and police image work : News media coverage of the rialto police department’s body worn camera experiment. *Crime, Media, Culture*, 14(3) :449–466.
- [Seigneur, 2006] Seigneur, E. (2006). *Etude et comparaison expérimentale de méthodes de localisation multicapteurs*. PhD thesis, Paris 11.
- [Servières et al., 2021] Servières, M., Renaudin, V., Dupuis, A., and Antigny, N. (2021). Visual and visual-inertial slam : State of the art, classification, and experimental benchmarking. *Journal of Sensors*, 2021 :1–26.
- [Shang et al., 2015] Shang, J., Hu, X., Gu, F., Wang, D., and Yu, S. (2015). Improvement schemes for indoor mobile location estimation : A survey. *Mathematical Problems in Engineering*, 2015.
- [Shin et al., 2007] Shin, S., Park, C., Kim, J., Hong, H., and Lee, J. (2007). Adaptive step length estimation algorithm using low-cost mems inertial sensors. In *2007 IEEE sensors applications symposium*, pages 1–5. IEEE.
- [Shipkovenski et al., 2020] Shipkovenski, G., Kalushkov, T., Petkov, E., and Angelov, V. (2020). A beacon-based indoor positioning system for location tracking of patients in a hospital. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–6.
- [Sikorska et al., 2011] Sikorska, J. Z., Hodkiewicz, M., and Ma, L. (2011). Prognostic modeling options for remaining useful life estimation by industry. *Mechanical systems and signal processing*, 25(5) :1803–1836.
- [Soudiene Mseddi et al., 2021] Soudiene Mseddi, W., Sedrine, M. A., and Attia, R. (2021). Yolov5 based visual localization for autonomous vehicles. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 746–750.
- [Strozzi et al., 2018] Strozzi, N., Parisi, F., and Ferrari, G. (2018). A novel step detection and step length estimation algorithm for hand-held smartphones. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE.
- [Stöckel et al., 2015] Stöckel, T., Jacksteit, R., Behrens, M., Skripitz, R., Bader, R., and Mau-Moeller, A. (2015). The mental representation of the human gait in young and older adults. *Frontiers in Psychology*, 6.
- [Sun et al., 2016] Sun, Y., Zhao, K., Wang, J., Li, W., Bai, G., and Zhang, N. (2016). Device-free human localization using panoramic camera and indoor map. In *2016 IEEE International conference on consumer electronics-China (ICCE-China)*, pages 1–5. IEEE.
- [Susi et al., 2013] Susi, M., Renaudin, V., and Lachapelle, G. (2013). Motion mode recognition and step detection algorithms for mobile phone users. *Sensors*, 13(2) :1539–1562.
- [Taia Alaoui, 2018] Taia Alaoui, F. (2018). *Navigation des personnes aux moyens des technologies des smartphones et des données d’environnements cartographiés*. PhD thesis, École centrale de Nantes.
- [Tan et al., 2020] Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet : Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.
- [Tateno et al., 2017] Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017). Cnn-slam : Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6243–6252.

- [Tian et al., 2019] Tian, Q., Kevin, I., Wang, K., and Salcic, Z. (2019). A resetting approach for ins and uwb sensor fusion using particle filter for pedestrian tracking. *IEEE Transactions on Instrumentation and Measurement*, 69(8) :5914–5921.
- [Tong et al., 2020] Tong, X., Su, Y., Li, Z., Si, C., Han, G., Ning, J., and Yang, F. (2020). A double-step unscented kalman filter and hmm-based zero-velocity update for pedestrian dead reckoning using mems sensors. *IEEE Transactions on Industrial Electronics*, 67(1) :581–591.
- [Valenti et al., 2015] Valenti, R. G., Dryanovski, I., and Xiao, J. (2015). Keeping a good attitude : A quaternion-based orientation filter for imus and margs. *Sensors*, 15(8) :19302–19330.
- [Varlamov and Aslamazov, 2012] Varlamov, A. and Aslamazov, L. (2012). *Wonders Of Physics, The (3rd Edition)*. World Scientific Publishing Company.
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [Vicon, 2023] Vicon (2023). <https://www.vicon.com>. consulté le 2023-05-01.
- [Wang et al., 2022] Wang, C., Zhang, Y., Zhou, Y., Sun, S., Zhang, H., and Wang, Y. (2022). Automatic detection of indoor occupancy based on improved yolov5 model. *Neural Computing and Applications*, pages 1–25.
- [Wang et al., 2021] Wang, Y., Cheng, H., Wang, C., and Meng, M. Q.-H. (2021). Pose-invariant inertial odometry for pedestrian localization. *IEEE Transactions on Instrumentation and Measurement*, 70 :1–12.
- [Wang et al., 2019] Wang, Z., Zhang, Q., Li, J., Zhang, S., and Liu, J. (2019). A computationally efficient semantic slam solution for dynamic scenes. *Remote Sensing*, 11(11) :1363.
- [Weinberg, 2002] Weinberg, H. (2002). Using the adxl202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note*, 2(2) :1–6.
- [Wen et al., 2023] Wen, S., Ge, Z., Yuan, D., Chen, Y., and Fang, X. (2023). Technology report : Smartphone-based pedestrian dead reckoning integrated with data-fusion-adopted visible light positioning. *arXiv preprint arXiv :2301.03471*.
- [Wojke et al., 2017] Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE.
- [Wu et al., 2018] Wu, J., Zhou, Z., Fourati, H., and Cheng, Y. (2018). A super fast attitude determination algorithm for consumer-level accelerometer and magnetometer. *IEEE Transactions on Consumer Electronics*, 64(3) :375–381.
- [Xiao et al., 2018] Xiao, A., Chen, R., Li, D., Chen, Y., and Wu, D. (2018). An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras. *Sensors*, 18(7) :2229.
- [Yan et al., 2019] Yan, H., Herath, S., and Furukawa, Y. (2019). Ronin : Robust neural inertial navigation in the wild : Benchmark, evaluations, and new methods. <http://ronin.cs.sfu.ca/>.
- [Yan et al., 2018] Yan, J., He, G., Basiri, A., and Hancock, C. (2018). Vision-aided indoor pedestrian dead reckoning. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6.

- [Yan et al., 2015] Yan, X., Liu, W., and Cui, X. (2015). Research and application of indoor guide based on mobile augmented reality system. In *2015 International Conference on Virtual Reality and Visualization (ICVRV)*, pages 308–311. IEEE.
- [Yao et al., 2021] Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J., and Li, X. (2021). A real-time detection algorithm for kiwifruit defects based on yolov5. *Electronics*, 10(14).
- [Yassin et al., 2017] Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E. (2017). Recent advances in indoor localization : A survey on theoretical approaches and applications. *IEEE Communications Surveys Tutorials*, 19(2) :1327–1346.
- [Ying et al., 2007] Ying, H., Silex, C., Schnitzer, A., Leonhardt, S., and Schiek, M. (2007). Automatic step detection in the accelerometer signal. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007) March 26–28, 2007 RWTH Aachen University, Germany*, pages 80–85. Springer.
- [Yu et al., 2017] Yu, C., El-Sheimy, N., Lan, H., and Liu, Z. (2017). Map-based indoor pedestrian navigation using an auxiliary particle filter. *Micromachines*, 8(7) :225.
- [Yu et al., 2019] Yu, H. W., Moon, J. Y., and Lee, B. H. (2019). A variational observation model of 3d object for probabilistic semantic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5866–5872.
- [Yuan et al., 2022] Yuan, H., Yu, H., Gui, S., and Ji, S. (2022). Explainability in graph neural networks : A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [Yun et al., 2008] Yun, X., Bachmann, E. R., and McGhee, R. B. (2008). A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *IEEE Transactions on instrumentation and measurement*, 57(3) :638–650.
- [Zafari et al., 2019] Zafari, F., Gkeliyas, A., and Leung, K. K. (2019). A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3) :2568–2599.
- [Zhang et al., 2019] Zhang, F., Wu, T.-Y., Pan, J.-S., Ding, G., and Li, Z. (2019). Human motion recognition based on svm in vr art media interaction environment. *Human-centric Computing and Information Sciences*, 9(1) :1–15.
- [Zhang et al., 2019] Zhang, J., Gui, M., Wang, Q., Liu, R., Xu, J., and Chen, S. (2019). Hierarchical topic model based object association for semantic slam. *IEEE Transactions on Visualization and Computer Graphics*, 25(11) :3052–3062.
- [Zhang et al., 2023] Zhang, J., Zhang, J., Zhou, K., Zhang, Y., Chen, H., and Yan, X. (2023). An improved yolov5-based underwater object-detection framework. *Sensors*, 23(7) :3693.
- [Zhang et al., 2022] Zhang, L., Bao, J., Xu, Y., Wang, Q., Xu, J., and Li, D. (2022). From coarse to fine : Two-stage indoor localization with multisensor fusion. *Tsinghua Science and Technology*, 28(3) :552–565.
- [Zhang et al., 2019] Zhang, W., Liu, G., and Tian, G. (2019). A coarse to fine indoor visual localization method using environmental semantic information. *IEEE Access*, 7 :21963–21970.
- [Zhao et al., 2019] Zhao, H., Cheng, W., Yang, N., Qiu, S., Wang, Z., and Wang, J. (2019). Smartphone-based 3d indoor pedestrian positioning through multi-modal data fusion. *Sensors*, 19(20) :4554.

- [Zhou et al., 2017] Zhou, Y., Zheng, X., Xiong, H., and Chen, R. (2017). Robust indoor mobile localization with a semantic augmented route network graph. *ISPRS International Journal of Geo-Information*, 6(7) :221.
- [Zhu et al., 2019] Zhu, J., Li, Q., Cao, R., Sun, K., Liu, T., Garibaldi, J. M., Li, Q., Liu, B., and Qiu, G. (2019). Indoor topological localization using a visual landmark sequence. *Remote Sensing*, 11(1).
- [Zhu et al., 2020a] Zhu, X., Qu, W., Qiu, T., Zhao, L., Atiquzzaman, M., and Wu, D. O. (2020a). Indoor intelligent fingerprint-based localization : Principles, approaches and challenges. *IEEE Communications Surveys & Tutorials*, 22(4) :2634–2657.
- [Zhu et al., 2020b] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020b). Deformable detr : Deformable transformers for end-to-end object detection. *arXiv preprint arXiv :2010.04159*.
- [Zijlstra and Hof, 1997] Zijlstra, W. and Hof, A. L. (1997). Displacement of the pelvis during human walking : experimental data and model predictions. *Gait & posture*, 6(3) :249–262.

Acronymes

A_{th} Seuil d'accélération. 66, 214

K_w Weiberg Gain. 213

SL step Length. 29, 67, 77, 78, 85, 213, 214, 217, 219

$d\theta$ Step Heading. 29, 30, 77, 78, 84, 85, 213, 214, 219

AAL *Ambient assisted living*. 71, 78

AP Point d'Accès. 35, 40

AR Réalité Augmentée. 7, 19, 22, 75, 113, 120, 121, 128

BIM *Building Information Model*. 93

BLE *Bluetooth Low Energy*. 35, 40

CDF Fonction de Distribution Cumulative. 6, 63, 73, 78

CNN Réseau de Neurones Convolutionnel. 114, 118, 122

COM centre de masse. 32, 66, 67

EKF filtre de Kalman étendu. 37, 38

GIS *Geographic Information System*. 93

GNSS *Global Navigation Satellite System*. 18, 35, 40, 74

GPU Unité de Traitement Graphique. 224

IA Intelligence Artificielle. 118, 121

IMMU Unité de Mesures Magnétiques et Inertielles. 31, 39, 44

IMU Unité de Mesures Inertielles. 30, 31, 32, 33, 34, 38, 39, 41, 44, 45, 46, 51, 52, 64, 75, 76, 115, 127, 160

IPIN *International Conference on Indoor Positioning and Indoor Navigation*. 71, 73, 76

KF filtre de Kalman. 37, 38, 39

MARG Magnetic, Angular Rate, and Gravity. 34

MEMS Système Micro-Électro-Mécanique. 45, 46, 61

MR Réalité Mixte. 7, 113, 120

PCD *Point Cloud Data*. 93

PDF fonction de Densité de Probabilité. 83, 84, 95

PDR navigation piétonne à l'estime. 5, 6, 11, 12, 22, 27, 28, 30, 31, 38, 39, 40, 41, 47, 63, 64, 66, 76, 77, 78, 107, 109, 110, 115, 214, 215, 216, 217, 219

PF filtre à particules. 12, 23, 37, 38, 39, 40, 41, 42, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 95, 96, 97, 99, 100, 102, 103, 104, 105, 107, 108, 109, 110, 130, 160, 161, 219, 220

RFID identification par radio-fréquence. 18, 35

RGB-D Red Green Blue - Depth. 117, 119, 127

RNG Random Number Generator. 99

RoNIN Robust Neural Inertial Navigation. 12, 31, 107, 109, 110

RSS Received Signal Strength. 40

RTK Real Time Kinetic. 74

SC Système de Coordonnées. 11, 47, 48, 50, 51

SCA Système de Coordonnées Appareil. 11, 49

SCG Système de Coordonnées Global. 11, 47, 49

SCU Système de Coordonnées Utilisateur. 47, 49

SLAM localisation et cartographie simultanées. 7, 22, 74, 75, 113, 117, 118, 119, 120, 127, 128

UKF filtre de Kalman non-parfumé. 37, 38, 120

UWB *Ultra Wide Band*. 35, 40

VLC Visible Light Communcation. 35

VR Réalité Virtuelle. 7, 19, 75, 113, 120, 128

WiFi Wireless Fidelity. 18, 35, 39, 40, 115, 127

Annexes

Annexe A

Acquisition de données sur Android

A.1 Disponibilité sur *GooglePlay*

<https://play.google.com/store/search?q=sensors>

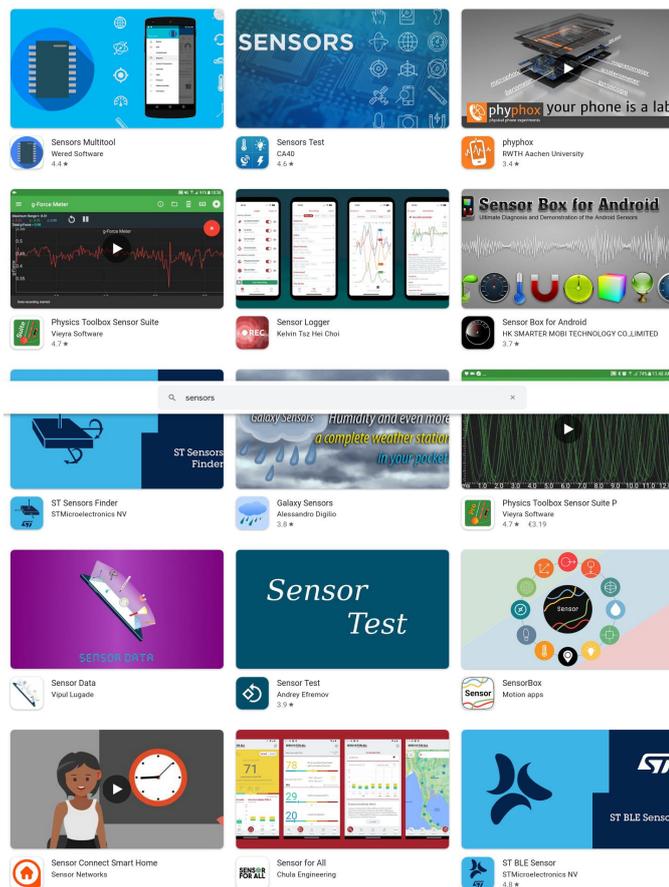


FIGURE A.1 – Résultat de la recherche du mot clef "sensors" sur *GooglePlay*.

Annexe B

Tests unitaires

B.1 Tests unitaires de l'estimation du déplacement

B.1.1 Test de détection de pas

Le but de ce test est de valider que la marche est bien un phénomène périodique discrétisable en une suite de pas détectables.

Protocole : l'utilisateur tient le téléphone verticalement, le port USB pointant vers le sol, l'écran face à lui. Il marche en comptant le nombre de pas qu'il fait.

Comportement attendu : l'utilisateur reporte avoir effectué 13 pas à environ 2 pas par secondes.

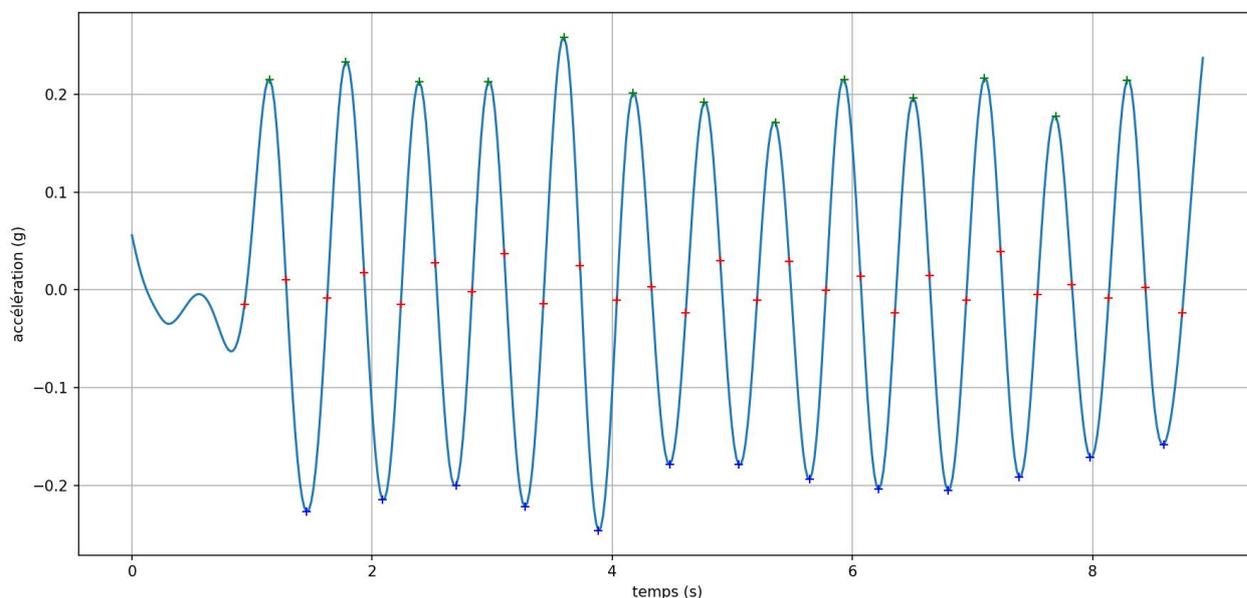


FIGURE B.1 – Visualisation de la détection de pas sur la norme d'accélération.

Résultat : la figure B.1 montre la courbe de la norme de l'accélération filtrée. Les points rouges indiquent les passages par zéro i.e. potentiellement le début, milieu ou la fin d'un pas. Les points rouges ne sont pas exactement alignés sur l'asymptote $y = 0$ car les échantillons sont

discrets. Les 13 paires de pics/vallées en verts/bleus correspondent aux maximum/minimum des 13 pas. L'enchaînement de la séquence : passage par zéro vers positif, pic, passage par zéro négatif, vallée puis passage par zéro positif correspond à un pas, il en existe exactement 13 dans cette figure. La figure montre aussi que le utilisateur effectue environ 2 pas par secondes. Le test de détection des pas est validé.

B.1.2 Test de passage sur les points repères

Le but de ce test est de montrer que l'utilisateur peut correctement déclarer le passage sur un point repère et que l'écart temporel entre l'instant choisi pour un pas i.e. l'atterrissage du pied, et l'instant de l'estampillage n'est pas significatif face à la durée d'un seul pas.

Protocole : l'utilisateur tient le téléphone verticalement, le port USB pointant vers le sol, l'écran face à lui. Il lui est demandé de piétiner une croix et d'appuyer au même moment sur le bouton.

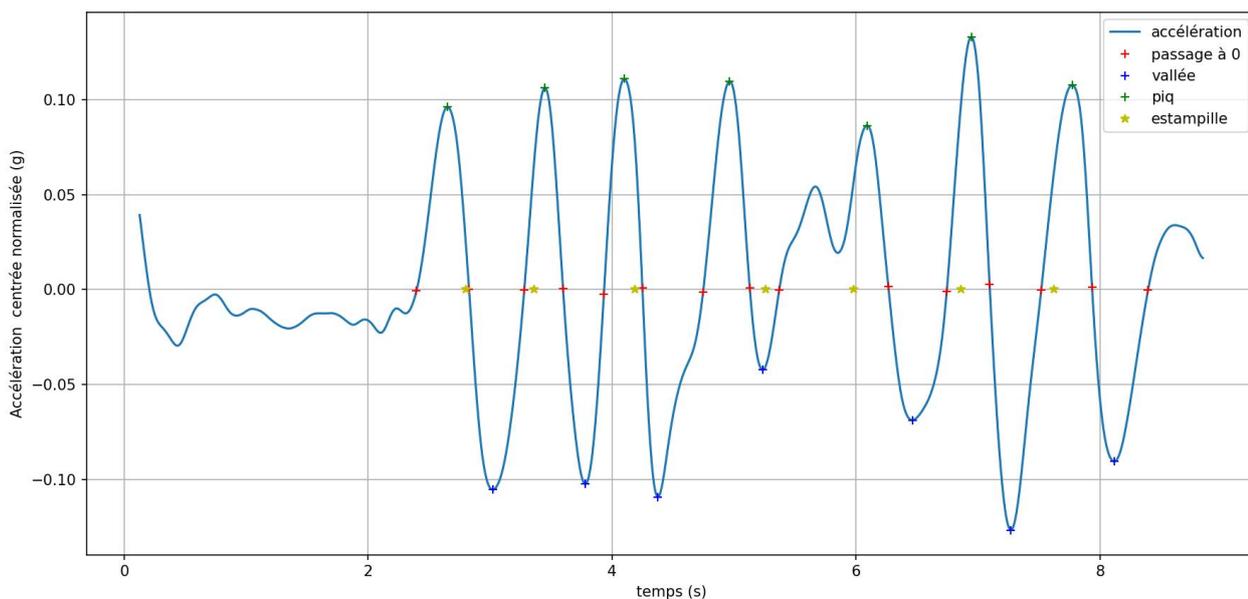


FIGURE B.2 – Sept pas effectuées sur points repères.

Comportement attendu : La détection de pas doit montrer la présence de 7 pas pour 7 croix. Il doit aussi donc y avoir 7 estampilles chacune incluse dans un pas unique.

Résultat : la figure B.2 montre la norme d'accélération filtrée avec 7 pas détectés. Les estampillés sont indiqués en avec des points jaunes. Il en existe 7 qui sont chacun inclus dans un pas unique. Cela montre bien que le piéton a réussi à déclarer son passage sur chaque croix. Toutefois, durant le test l'utilisateur a déclaré qu'exécuter les deux actions d'activation piétiner une croix et d'appui sur un bouton pouvait provoquer une confusion. Cela se voit à $t = 5.75s$ où il existe 2 pics où seul l'un des deux est un pic de pas. Donc, pour une trajectoire expérimentale, les points repères doivent être assez espacés entre eux.

B.1.3 Test de distance parcourue

Ce test a pour but d'estimer la distance parcourue par un utilisateur et ensuite d'estimer le gain de *Weinberg* lié à celui ci.

Protocole : l'utilisateur tient le téléphone verticalement, le port USB pointant vers le sol, l'écran face à lui. Il commence en position d'arrêt avec les pieds joints. L'acquisition est lancée et le top est donné. L'utilisateur marche tout droit à son grès. On lui demande de s'arrêter avec les pieds joints. La distance parcourue est mesurée avec un télémètre laser.

Comportement attendu : L'utilisateur a marché une distance mesurée qui est comparée à la distance estimée par la méthode de *Weinberg*.

Résultat : la figure B.1 montre l'évolution de la norme d'accélération durant cette marche. En utilisant la méthode Weinberg, la distance parcourue à chaque pas peut être calculée avec les pics (verts) et vallées (bleus).

On rappelle que K_w peut être calculé tel que :

$$K_w = \frac{D_{\text{mesurée}}}{D_{\text{estimée}}} \quad (\text{B.1})$$

Où $D_{\text{mesurée}}$ est donnée par le télémètre laser et $D_{\text{estimée}}$ est calculée en posant d'abord K_w à 1 dans l'équation (4.6) rappelée ci-dessous :

$$D_{\text{estimée}} = \sum_{i=1}^P K_w \times \sqrt[4]{A_{\text{max}} - A_{\text{min}}} \quad (\text{B.2})$$

B.1.4 Test de trajectoire rectiligne

Ce test vise à recréer la trajectoire dans un couloir pour montrer que la combinaison de la longueur du pas SL et de la différence d'angle $d\theta$ est possible.

Protocole : le téléphone est attaché au torse du téléphone du piéton à la verticale, le port USB pointant vers le sol. Le piéton marche d'un point de départ à un point d'arrivée situé 15 m plus loin.

Comportement attendu : La trajectoire a une allure rectiligne et sa longueur est proche de 10 m.

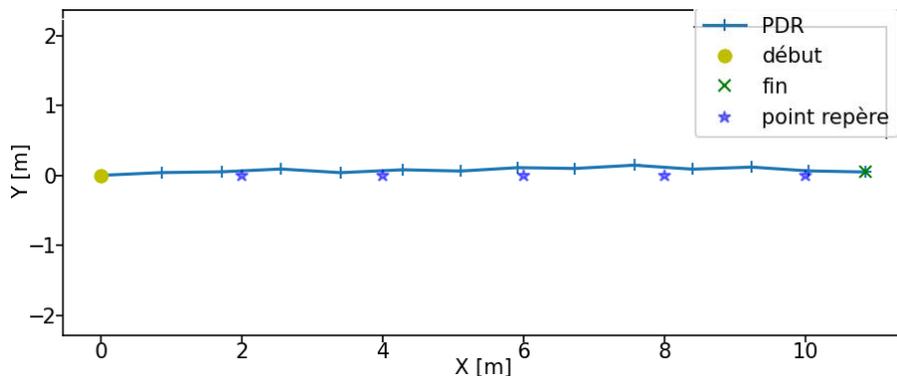


FIGURE B.3 – Trajectoire rectiligne visualisée.

Résultat : la figure B.3 montre que la trajectoire est bien rectiligne et qu'elle est composée d'une succession de pas droits et gauches. La distance estimée est de 10.81 m soit un erreur

euclidienne de 0.84 m. Il y a donc une surestimation de la longueur du pas. Ce test est répétable et parfois donne une sous-estimation de la distance parcourue même si K_w est estimé. Ceci implique qu'il faut un mécanisme de correction de la longueur de pas qui est présenté dans le chapitre 5 rejoignant le constant de l'état de l'art du chapitre 2. Le test est validé.

B.1.5 Test de l'invariance aux utilisateurs

Le but de ce test est d'étudier l'effet des deux paramètres : le gain de *Weinberg* K_w et le seuil d'accélération A_{th} sur la PDR.

Protocole : chacun des utilisateurs marche jusqu'à ce que l'arrêt lui soit demandé. La distance parcourue est mesurée au télémètre et K_w est estimée pour chaque utilisateur.

Comportement attendu : pour les paramètres estimées, la distance estimée est proche de la distance réelle. Un A_{th} produit des faux positifs/négatifs qui induit une erreur importante. La permutation de paramètres inter-utilisateurs fausse totalement les prédictions de distance.

Utilisateur	Sexe	Taille	# de pas	K_w	A_{th}
1	M	1m90	13	1.01169	0.03
2	M	1m86	14	1.00169	0.03
3	M	1m80	16	0.98505	0.03
4	M	1m70	18	0.96123	0.03
5	M	1m64	21	0.94691	0.03

TABLE B.1 – Données prélevées et calculées sur différents utilisateurs.

Résultat : le tableau B.1 montre que K_w est bien proportionnel à la taille de l'utilisateur qui est elle même proportionnelle à la longueur de sa jambe. Mais, ce n'est pas le cas pour A_{th} . Le seuil dépend de la cadence et la nature de marche. Une cadence plus élevée ou une marche "stressée" montre des pics ou vallées qui ne résultent pas uniquement du pas.

B.1.6 Test de trajectoire curviligne

Ce test vise à valider la combinaison de SL et $d\theta$ quand la trajectoire est curviligne i.e. un virage.

Protocole : Dans les mêmes conditions que le test précédent, le piéton effectue un virage vers sa gauche et s'arrête. Le coin autour duquel le virage est formé est positionné de manière à ce que le point d'arrivée soit à une translation de (1.5, 1.5).

Comportement attendu : La trajectoire aura une allure curviligne. Le point de départ est aux coordonnées (0, 0) le point d'arrivée est aux coordonnées (1.5, 1.5). Le virage est pris vers la gauche du piéton.

Résultat : la figure B.4 montre que la trajectoire est bien curviligne et a suivi l'allure du virage pris vers la gauche. La courbe est composée de segment discrets pour chaque pas. Le point d'arrivée est aux coordonnées (1.55, 1.70) ce qui implique qu'il existe une erreur de 0.15 m liée soit à l'estimation de SL , $d\theta$ ou des deux. Le test est validé.

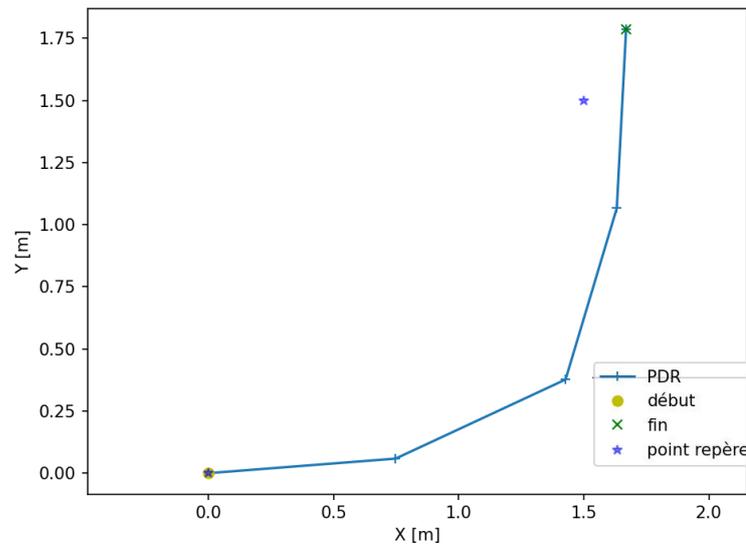


FIGURE B.4 – Trajectoire curviligne visualisée.

B.1.7 Test de trajectoires par navigation à l'estime

Ce test vise à estimer par la PDR les trajectoires que prend l'utilisateur : un aller-retour et un carré. Ces trajectoires combinent les trois aspects précédents : détection d'un pas, estimation de sa longueur et de son cap. Ils représentent un premier résultat exploitable par l'utilisateur.

La trajectoire présentée en figure B.5 n'a une allure d'un aller-retour. Pour faire demi tour, le piéton ne s'arrête pas pour tourner sur place (rotation de tourelle). Il effectue le demi tour via une combinaison de tourner son corps et translater son pied vers l'arrière en même temps. Donc, l'écart entre la direction de marche et le cap du corps est important. Il faut au moins 2 pas pour faire ce mouvement. Ceci implique que le demi tour est un mode déplacement distinguable dans la marche. Il est détectable par une différence de cap entre pas proche de 180° sur 2 ou 3 pas.

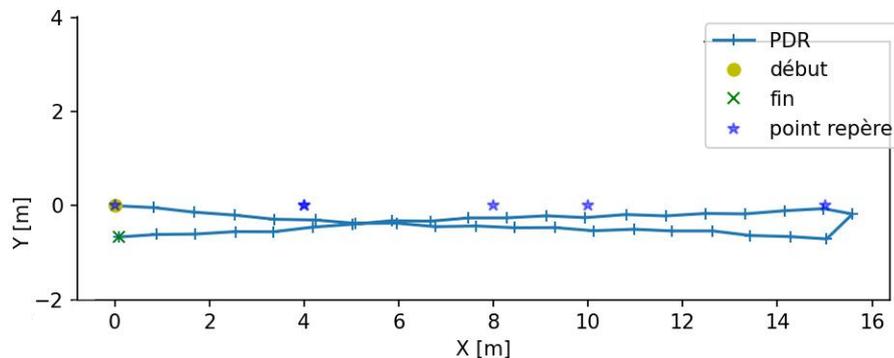


FIGURE B.5 – Trajectoire aller-retour visualisée.

La trajectoire de la figure B.6 présente le résultat de la PDR appliquée aux données d'une trajectoire carrée faisant $15\text{ m} \times 15\text{ m}$. La trajectoire a bien une allure carrée avec 4 côtés de longueurs proche de 15 m. L'erreur euclidienne est de 1.33 m. La trajectoire est supposée faire

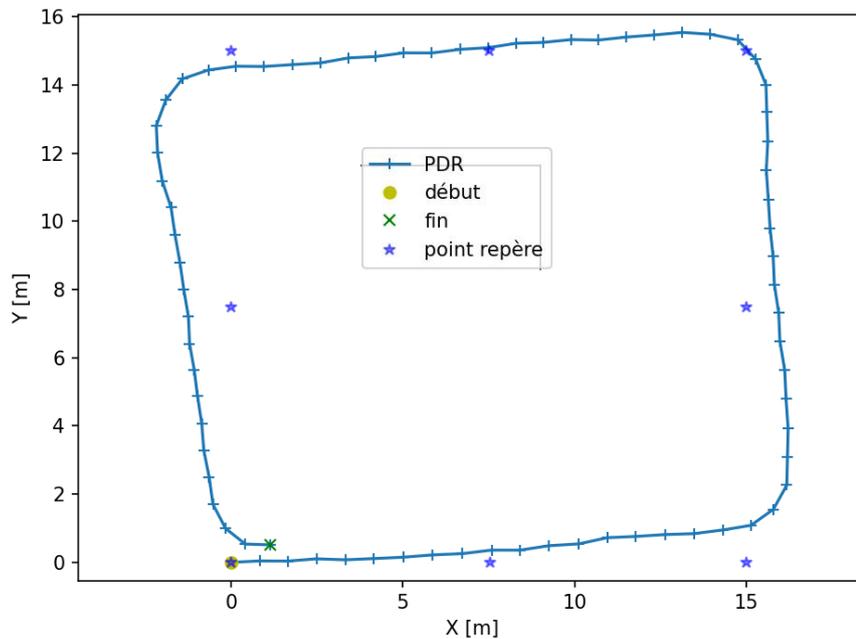


FIGURE B.6 – Trajectoire carrée visualisée.

le tour de quatre murs mais la PDR n’incorpore pas cette information. Il est donc possible que la trajectoire estimée indique que le piéton traverse un mur.

B.1.8 Test de l’orientation en 3D

Dans la section 3.3, on émet l’hypothèse que le téléphone est maintenu dans la même position et orientation que le piéton.

Protocole : Afin de vérifier l’impact de cette hypothèse, on procède à l’acquisition d’une séquence de données sur une trajectoire en boucle sur un seul carré. Puis, on visualise l’évolution des 3 angles d’orientation calculés par le filtre de *Madgwick* durant toute la séquence.

Comportement attendu : L’angle de roulis (ϕ) devrait avoir une moyenne de 90° avec une variance correspondant au bruit d’acquisition. 90° est la valeur pour laquelle le repère du filtre de *Madgwick* indique que le téléphone est tenu en mode portrait.

L’angle de tangage (ψ) a une moyenne nulle avec une variance correspondant au bruit d’acquisition si le téléphone est bien vertical par rapport au plan du sol.

L’angle du cap (α) suit les virages que prend le piéton. Il y a autant de rotation à 90° que de virages dans la trajectoire.

Résultat : La figure B.7 représente l’évolution des 3 angles concernés. On remarque d’abord que l’angle α représente bien les virages à 90° de la trajectoire carrée autour de 9, 18, 23, 27, 32, 36,... secondes.

Mais, on remarque aussi que les angles ϕ et ψ ne sont pas aux valeurs attendues avant 20 s. Ceci impliquerait que le filtre de *Madgwick* a un temps de convergence de 20 s. Plus précisément, il converge en 20 s à 50 Hz. Il faut donc 1000 échantillons avant de converger.

On relève dans le tableau B.2 les moyennes (μ) et écart-type (σ) dans les intervalle : [0-20], [20-110] et [0-110] secondes pour ϕ et ψ .

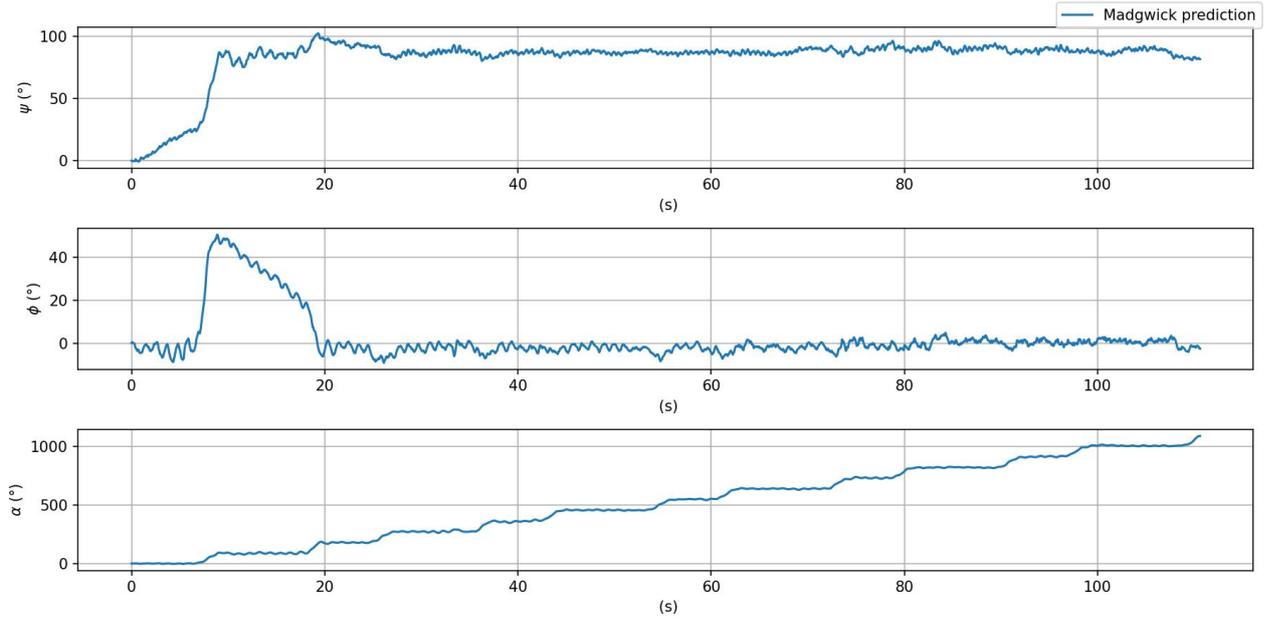


FIGURE B.7 – Évolution des angles d’orientation prédits par le filtre de Madgwick.

temps [s]	$\mu_\psi [^\circ]$	$\sigma_\psi [^\circ]$	$\mu_\phi [^\circ]$	$\sigma_\phi [^\circ]$
[0 – 110]	82.598	19.333	2.1489	11.445
[0 – 20]	57.192	35.402	18.478	19.488
[20 – 110]	88.171	2.854	-1.4331	2.2562

TABLE B.2 – Moyennes et écart-types des angles ψ et ϕ pour la figure B.7.

Il est donc nécessaire de corriger le cap estimé par le filtre de *Madgwick*. Nous présentons la correction de ce cap via l’ajout de la carte du bâtiment dans le chapitre 5.

B.1.9 Test de trajectoire par navigation à l’estime 3D

Protocole : Le test précédent a montré que le filtre de *Madgwick* peut aussi suivre correctement les angles de roulis et de tangage après un temps de convergence avec une erreur acceptable. On souhaite connaître l’impact de cette erreur sur une trajectoire en 3D. L’utilisateur marche suivant une trajectoire carrée en tenant le téléphone dans les mêmes conditions que le test précédent.

Dans le cas 3D, la PDR passe par une représentation en coordonnées sphériques : la longueur de pas SL , et les angles représentés sur la figure B.8 à l’instant t . Le triplet de position (x, y, z) à l’instant d’un pas t est calculé par les équations (B.3) à (B.5).

$$x_t = \cos \alpha_t \times \sin \beta_t \times SL_t + x_{t-1} \quad (\text{B.3})$$

$$y_t = \sin \alpha_t \times \sin \beta_t \times SL_t + y_{t-1} \quad (\text{B.4})$$

$$z_t = \cos \beta_t \times SL_t + z_{t-1} \quad (\text{B.5})$$

$$\beta_t = \pi/2 - \psi_t \quad (\text{B.6})$$

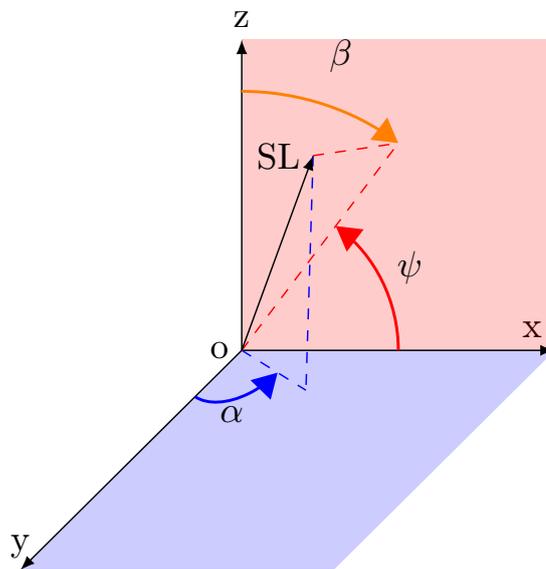


FIGURE B.8 – Système de coordonnées sphériques.

Comportement attendu : la trajectoire a une allure carrée. La valeur sur l'axe de la hauteur (z) indique à quel point la dérive en z affecte l'estimation des déplacements. Cette valeur en z est très petite comparée à l'erreur de localisation car le piéton ne prend pas d'escaliers, ne se baisse pas et n'est pas dans un ascenseur.

Les valeurs des positions en x et y correspondent à celles présentées sur la figure B.6.

Les figure B.9 ont une allure correcte par rapport aux trajectoires carrées demandées. On remarque une dérive quasi-constante sur l'axe z . Depuis le tableau B.2, nous avons noté que ψ n'était pas exactement nul et donc β ne l'est pas non plus (équation (B.6)). Ceci fait que la composante z de la position accumule une distance proportionnelle à $\cos \beta$ à chaque pas. Ce phénomène ne cumule pas la dérive que dans un seul sens de z comme montré la figure B.9 à droite. Il réduit aussi les composants x et y de la position car $\sin \beta \in [0, 1]$ même s'il reste proche de 1.

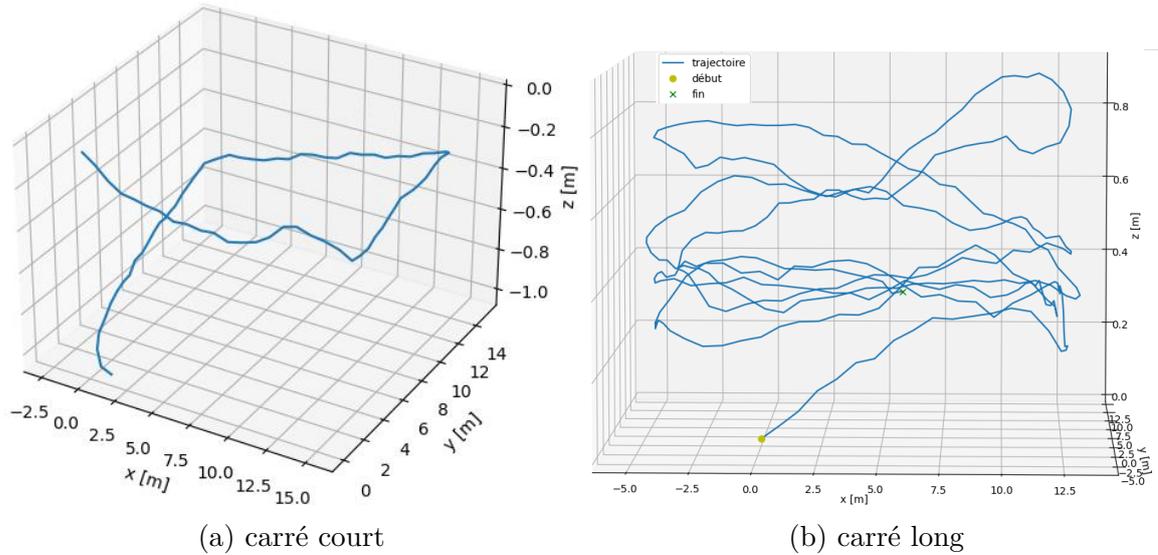


FIGURE B.9 – Visualisation de trajectoires en 3D.

B.2 Tests unitaires du filtrage particulaire

B.2.1 Test unitaire du filtrage par murs

Dans ce test, un pas d'une longueur dans une direction connue est simulé en ajoutant les contraintes liées aux murs présents dans le bâtiment. Le but est de vérifier que seules les particules valides se trouvent dans le support attendu.

La figure B.10 montre un exemple d'un filtrage par les murs pour une prédiction SL de 3 m et $d\theta = 0$ rad. Les segments bleus délimitent le support (espace des hypothèses restreint) après la prédiction de leurs nouvelles positions et les particules invalides ne sont pas affichées. Le filtrage des particules par les murs de la carte pour un pas est validé.

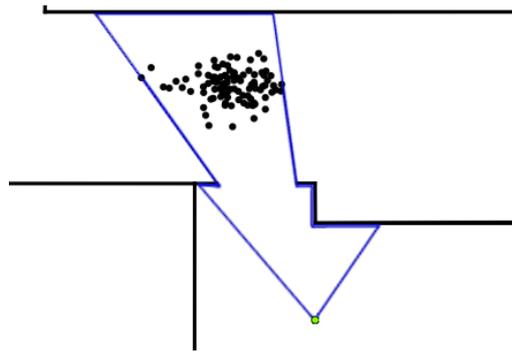


FIGURE B.10 – Filtrage des particules par les murs.

B.2.2 Trajectoire carrée filtrée et régénérée

Pour ce test, le piéton effectue une trajectoire autour d'un carré. Pour chaque pas, on enchaîne l'estimation par la PDR puis estimation par PF. On représente uniquement les murs autour desquels le piéton a fait la trajectoire comme un carré (4 segments). Dans un premier

temps les particules prédites traversant le mur sont éliminées. Dans un second temps, la régénération est activée pour permettre au PF d'avoir les N particules nécessaires à son fonctionnement. Dans les deux cas le piéton est sensé être revenu à sa position de départ. À noter que pour cette expérience le tirage des nombres aléatoires est figé et identique pour les deux cas pour ne pas biaiser l'expérience.

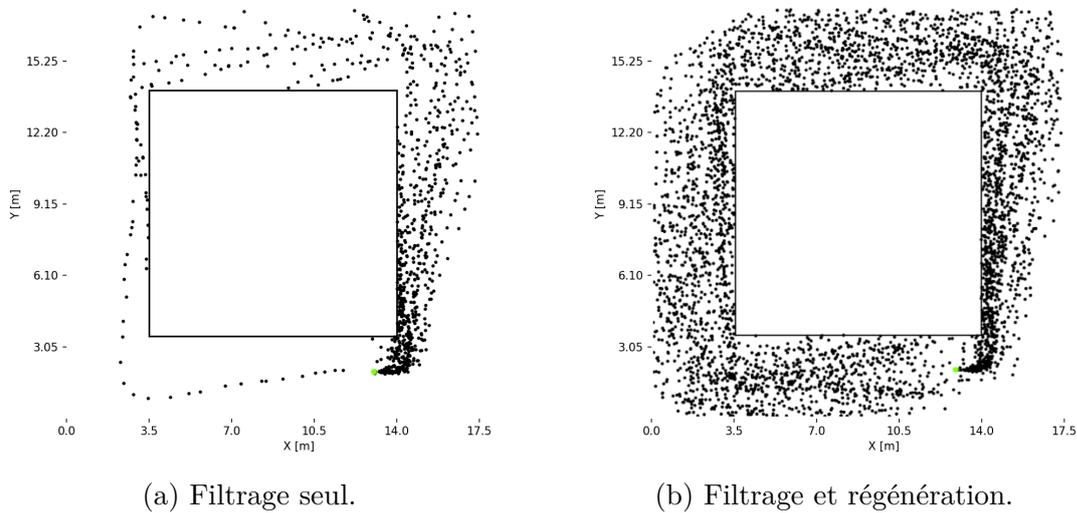


FIGURE B.11 – Propagation de 50 particules visualisée sur une trajectoire carrée avec :

Dans la figure B.11a, quand seul le filtrage par mur est actif, le nombre de particules diminue progressivement tout le long de la trajectoire jusqu'à n'avoir qu'une seule particule à la fin. Le filtre est appauvri et perd tout robustesse et ne représente plus l'espace des hypothèses. Il est possible que la localisation échoue si toutes les particules sont détruites.

Une fois la régénération activée, le filtre conserve N particules comme le montre la figure B.11b. Les particules sont disposées de manière relativement proche de la position de départ. La combinaison filtrage par mur et régénération est donc nécessaire au bon fonctionnement du filtre.

Calcul dans le filtre particulière

B.3 Adjacence et norme de distance

Après l'étape de discrétisation de la carte en grille, une carte des distances de chaque case au mur le plus proche est dressée. Cette carte des distances peut être calculée avec 3 normes de distance :

Euclidienne : chaque case calcule la distance euclidienne entre le mur le plus proche et elle-même. Cela implique qu'il faut une recherche de ce mur dans un voisinage autour d'une case en agrandissant ce voisinage jusqu'à trouver un mur. Il suffit de sauvegarder la distance euclidienne entre le premier mur trouvé et la cellule actuelle. Après implantation, ce processus de construction a été jugé trop long pour une carte de $36\text{ m} \times 70\text{ m}$. Ce qui fait 1 008 000 de cases à calculer pour un étage. Même si la grille n'est calculée qu'une seule fois, l'usage d'une norme de distance plus rapide est nécessaire dans le cas où l'utilisateur explore un bâtiment et que la grille n'a jamais été construite auparavant.

8-adjacence la norme de distance entre une case p et q au indices (i_p, j_p) et (i_q, j_q) en 8-adjacence est donnée par :

$$d = \min(|i_p - i_q|, |j_p - j_q|) \quad (\text{B.7})$$

Donc une cellule considère ces 8 voisins cardinaux lors de la détection du mur.

4-adjacence la norme de distance entre une case p et q ayant les indices (i_p, j_p) et (i_q, j_q) en 4-adjacence est donnée par :

$$d = |i_p - i_q| + |j_p - j_q| \quad (\text{B.8})$$

Donc une cellule considère seulement ces 4 voisins cardinaux lors de la détection du mur.

Construction Une méthode simple pour construire la grille en utilisant soit la 4 ou la 8-adjacence a été proposée par [Felzenszwalb and Huttenlocher, 2012]. En notant Nord N, Ouest O, Est E et Sud S, elle est résumée par les étapes suivantes :

1. Initialiser toutes les cases à ∞
2. Initialiser les murs à 0.

3. Balayer du NW vers SE :

Pour 4-adjacence, pour une cellule c , calculer

$$c(i, j) = \min(d(i, j), d(i-1, j) + 1, d(i, j-1) + 1) \quad (\text{B.9})$$

Pour 8-adjacence, pour une cellule d , calculer

$$c(i, j) = \min(d(i, j), d(i-1, j-1)+1, d(i-1, j)+1, d(i, j-1)+1, d(i-1, j+1)+1) \quad (\text{B.10})$$

4. Balayer du NW vers SE : en balayant du SE vers NW.

Pour 4-adjacence, pour une cellule d , calculer

$$c(i, j) = \min(d(i, j), d(i+1, j) + 1, d(i, j+1) + 1) \quad (\text{B.11})$$

Pour 8-adjacence, pour une cellule d , calculer

$$c(i, j) = \min(d(i, j), d(i+1, j+1)+1, d(i+1, j)+1, d(i, j+1)+1, d(i+1, j-1)+1) \quad (\text{B.12})$$

B.4 Algorithmes de ré-échantillonnage

La figure B.12 donne une visualisation de cinq de ces algorithmes de ré-échantillonnage. Pour chacun d'entre eux, on considère le même vecteur de 10 poids en entrée :

$$\mathbf{w} = 0.172, 0.001, 0.181, 0.007, 0.154, 0.037, 0.182, 0.114, 0.063, 0.089 \quad (\text{B.13})$$

À chaque fois on liste, les étapes de l'algorithme. [Douc and Cappe, 2005] analyse avec plus de détails ces algorithmes dans la section 2 de sa publication.

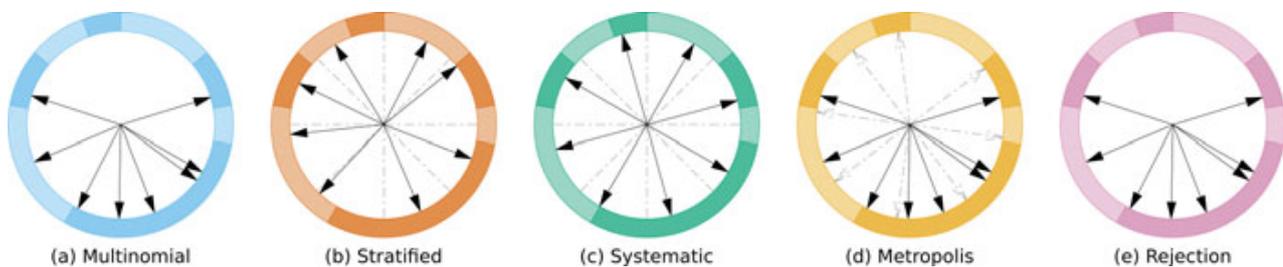


FIGURE B.12 – Visualisation de 5 algorithmes de ré-échantillonnage. Image de [Murray et al., 2016].

Multinomial :

1. Tirer $\mathbf{u} \sim U[0, 1]^N$, N valeurs tirées uniformément et ordonnées.
2. Calculer la somme cumulée de \mathbf{w} :

$$\mathbf{cs} = \left\{ \sum_{j=1}^i w_j \right\}_{i=1}^N \quad (\text{B.14})$$

3. Pour chaque w^i , garder la particule P^i si :

$$cs_{i-1} \leq u_i < cs_i \quad | \quad cs_0 = 0 \text{ et } cs_N = 1 \quad (\text{B.15})$$

4. Remplir les indices manquants en dupliquant le voisin précédent.

Exécution :

Entrée	w	=	0.172	0.001	0.181	0.007	0.154	0.037	0.182	0.114	0.063	0.089	
Tirage	u	=	0.017	0.041	0.270	0.544	0.607	0.637	0.729	0.813	0.913	0.935	
Cumul	cs	=	0	0.172	0.173	0.354	0.361	0.515	0.552	0.734	0.848	0.911	1.0
Test	o	=	0		2				6	7		9	
Remplissage	o	=	0	0	2	2	2	2	6	7	7	9	

Stratifié :

1. Découper l'intervalle $(0, 1]$ en N ensembles disjoints de taille égale notés μ_k .
2. Tirer aléatoirement uniformément **u** :
> chaque élément u_k est tiré indépendamment dans chaque μ_k .
3. Procéder à un ré-échantillonnage multinomial avec les nombres tirés **u**.

Systematique :

1. Reprendre quasiment les mêmes étapes que la version stratifiée.
2. Sauf qu'il faut tirer un nombre aléatoire $a \sim U[0, 1]$.
3. Et calculer déterministiquement **u** tel que :

$$u_i = (i - 1)/N + a \quad | \quad i \in \{1, \dots, N\} \quad (\text{B.16})$$

4. Procéder au ré-échantillonnage multinomial.

Jusque là, tous les algorithmes vus précédemment ont une dépendance de données inter-poids i.e. la somme cumulée des poids est une étape faiblement parallélisable. Les deux algorithmes suivants ont pour particularité de surmonter ce problème.

Metropolis : Cet algorithme proposé par l'auteur éponyme [Metropolis et al., 1953] est parallélisable pour chaque particule si une copie des poids est faite au préalable. Les opérations suivantes sont faites par chaque particule E^i .

1. Initialiser chaque k à i .
2. Faire un tirage uniforme d'un nombre aléatoire $u \sim U[0, 1]$.
3. Tirer uniformément un indice $j \sim U\{1, \dots, N\}$.
4. Si $u < w^j/w^k$.
> k reçoit j .
5. Répéter B fois depuis l'étape 2.
6. E^i devient E^k .

L'un des avantages et inconvénients que présente cet algorithme est la présence du paramètre de B . Une valeur faible de B rend l'algorithme plus rapide mais produit un résultat biaisé. B représente alors un compromis vitesse-biais qui dépend fortement des applications de filtrage selon [Murray et al., 2016].

Réjection : Si une borne supérieure des poids w_{max} est connue *a priori*, le ré-échantillonnage par réjection est possible [Murray et al., 2016, Douc and Cappe, 2005]. Celui-ci a les mêmes avantages que *Metropolis* mais en plus, il n'est pas biaisé et permet de réduire la variance de ré-échantillonnage. Donc pour chaque particule E^i , les opérations suivantes sont effectuées :

1. j reçoit i .
2. Tirer $u \sim U[0, 1]$.
3. Tant que $u > w^j/w_{max}$.
 $j \sim U\{1, \dots, N\}$.
 $u \sim U[0, 1]$.
4. E^i devient E^j .

Un inconvénient unique lié à cet algorithme (et à son nom) est que la réjection à l'étape 3 peut demander un temps d'exécution assez important jusqu'à l'acceptation. Ceci est important à considérer dans le cas où le parallélisme est exécuté sur GPU. Il peut y avoir un *thread* bloquant tout un bloc [Murray et al., 2016, Murray, 2012].

Résiduel : La sixième famille est celle basée sur l'algorithme **résiduel**. Cet algorithme vise à réduire la variance du nombre de copies de chaque particule dupliquée.

1. Créer $\lfloor Nw^i \rfloor$ copies d'une particule E_i .
2. Étant donné que $\lfloor Nw^i \rfloor \leq N$, il faut ajouter \bar{N} tel que :

$$\bar{N} = \sum_{i=1}^N Nw^i - \lfloor Nw^i \rfloor \quad (\text{B.17})$$

3. Effectuer un ré-échantillonnage multinomiale sur les \bar{N} particules dont le poids sera calculé tel que :

$$w^i = (Nw^i - \lfloor Nw^i \rfloor) / \bar{N} \quad (\text{B.18})$$

Annexe C

Vision par caméra

C.1 Modèle de caméra sténopé

L'appareil photo du téléphone est représenté comme un modèle sténopé. Cela signifie que les coordonnées spatiales 3D (x, y, z) sont projetées dans l'espace 2D représenté dans l'image par les coordonnées pixel (u, v) et S étant l'incertitude liée à la projection. Les deux espaces sont liés par l'équation (C.1) exprimée en coordonnées homogènes. Il faut noter que selon la perspective de la caméra sur la figure C.1 : x est la profondeur, y est projeté sur les colonnes de l'image et z est projeté sur les lignes. C'est pourquoi le vecteur a l'ordre suivant : $[y \ z \ x \ 1]^T$. La profondeur x est perdue lors de la projection. u dans l'image évolue dans le sens opposé à y dans l'espace 3D. Aller dans les colonnes de droite de l'image fait évoluer positivement l'angle dans l'image. Dans l'espace 3D, l'angle évolue positivement en partant de 0° (Est) dans le sens antihoraire.

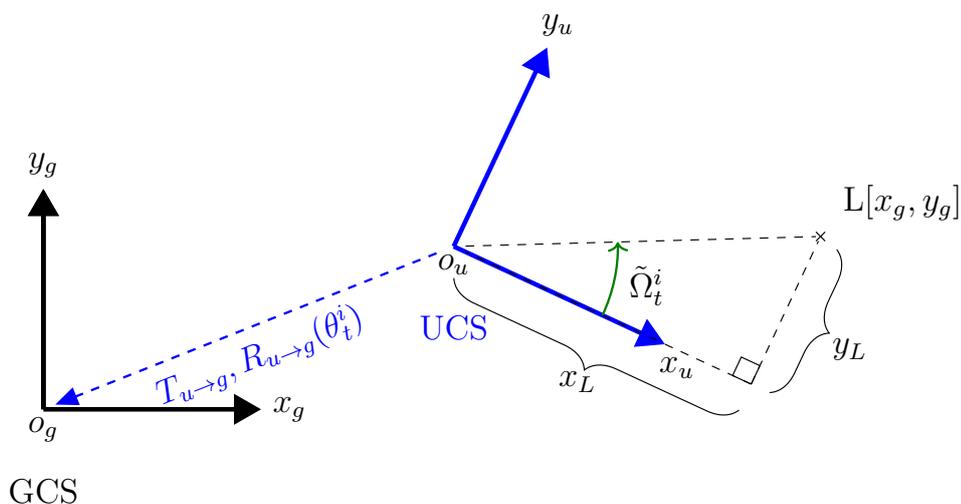


FIGURE C.1 – Vue à vol d'oiseau des systèmes de coordonnées de la figure 8.3.

$$\begin{bmatrix} Su \\ Sv \\ S \end{bmatrix} = \begin{bmatrix} -F_u & 0 & C_u & 0 \\ 0 & F_v & C_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} y \\ z \\ x \\ 1 \end{bmatrix} \quad (\text{C.1})$$

$$\Rightarrow = \begin{bmatrix} -F_{u_y} + C_{u_x} \\ F_{v_z} + C_{v_x} \\ x \end{bmatrix} \quad (\text{C.2})$$

$$\Rightarrow \begin{cases} y_g^O = \frac{-(Su - C_u x_g^O)}{F_u} \\ x_g^O = S \end{cases} \quad (\text{C.3})$$

Pour un point de repère de coordonnées (x_g^O, y_g^O, z_g^O) apparaissant dans la caméra, $x = x_g^O, y = y_g^O$. Dans l'équation (C.3), en remplaçant x dans l'expression de y et en calculant y/x on obtient :

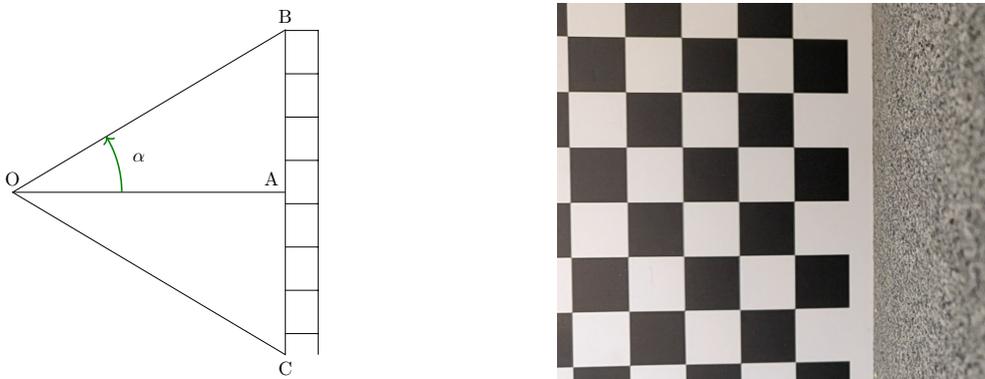
$$\Omega_t^O = -\arctan \frac{y_g^O}{x_g^O} = -\arctan \frac{u - C_u}{F_u} \quad (\text{C.4})$$

Cela rend le calcul de Ω_t^O indépendant de la position du point de repère. Cela ne dépend que de u qui est l'indice de colonne de pixels dans l'image, C_u et F_u étant des constantes de la matrice de calibrage.

Remarque : Le système de coordonnées de la caméra considère que les colonnes évoluent de gauche à droite. Dans la figure 8.3, l'axe de la colonne de la caméra, y_c est de direction opposée à y_u d'où l'apparition du signe moins dans l'équation (C.4).

$$R_{c \rightarrow p \rightarrow g} = \begin{bmatrix} \cos(\theta_t^i) & \sin(\theta_t^i) \\ -\sin(\theta_t^i) & \cos(\theta_t^i) \end{bmatrix} \quad (\text{C.5})$$

L'angle d'ouverture de la caméra est obtenu expérimentalement en la plaçant face à un motif en damier à une distance connue. Dans la figure C.2b, on peut voir 7 carrés et demi. Dans la figure C.3, le damier comporte 14 cases d'une longueur totale de 0,563 m.



(a) Vue à vol d'oiseau de la figure C.2b. (b) Exemple de carrés pouvant être vus dans le cadre

FIGURE C.2 – Damier de calibration.

L'angle d'ouverture peut être calculé comme suit :

$$\alpha = \arctan \frac{\vec{AB}}{\vec{AO}} \quad (\text{C.6})$$

avec :

1. $AO = 0,5 \text{ m}$
2. $AB = \frac{7,5 \times 0,563}{14 \times 2} \text{ m}$

En suivant le guide OpenCV pour l'étalonnage¹, un damier (figure C.3) est utilisé pour obtenir la matrice d'étalonnage de la caméra sténopé. En effectuant divers mouvements en mode vidéo, 460 images sont acquises puis soumises au processus de calibrage.

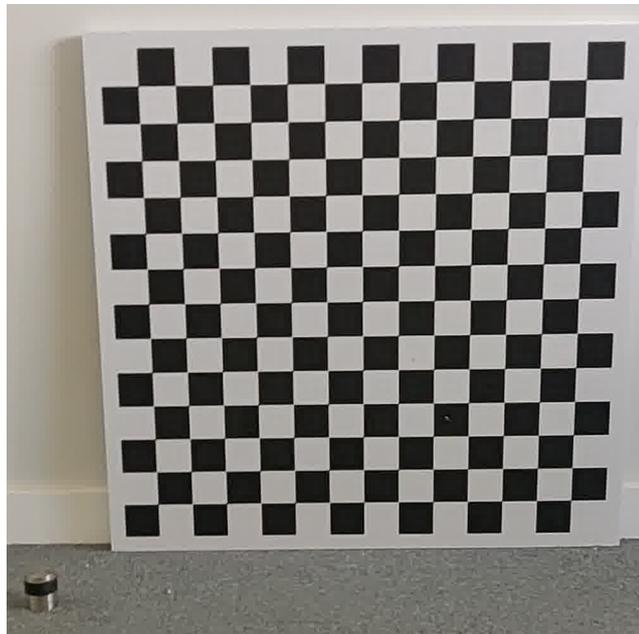


FIGURE C.3 – Motif de damier.

Pour valider les valeurs expérimentales trouvées par étalonnage, un repère virtuel est créé pour imiter la sortie de YOLOv5 :

label xc yc dx dy c

1. label : la classe à laquelle appartient l'objet détecté
2. xc : [0,1], le centre d'index de colonne de l'objet par rapport à la largeur de l'image.
3. yc : [0,1], le centre de l'index de ligne de l'objet par rapport à la hauteur de l'image.
4. dx : [0,1], la largeur relative de l'objet.
5. dy : [0,1], la hauteur relative de l'objet.
6. c : [0,1] confiance dans la détection.

1. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

En mettant x_c à 0 ou 1, le centre de l'objet est placé respectivement à gauche ou à droite de l'image. Cela signifie que la valeur de l'angle sous lequel la caméra voit le repère artificiel est à sa valeur minimale ou maximale. Après test, les résultats étaient :

1. $x_c=0$; 17°
2. $x_c=1$; 14.4°

Il existe une différence entre l'utilisation du mode vidéo du téléphone et celle de l'appareil photo en raison d'Android qui utilise différentes caméras pour chaque mode. L'angle d'ouverture du mode photo est de $\approx \pm 22,5^\circ$ tandis que celui du mode vidéo est de $\approx \pm 15,5^\circ$ Pour le téléphone *OnePlus 6T*. Android a la mise au point automatique continue et l'exposition automatique activées par défaut. C'est pourquoi les paramètres intrinsèques F_u, F_v évoluent dans le temps. L'utilisation d'une application pour les verrouiller est obligatoire.

Étalonnage distance - détection On souhaite construire la fonction d'étalonnage f défini par :

$$f : du \in]0, 1] \subset \mathbb{R} \rightarrow d \in \mathbb{R}_*^+ \quad (\text{C.7})$$

Pour cela, on suit ce protocole :

1. Un extincteur est posé au sol contre un mur comme dans la figure C.4.
2. On mesure au télémètre laser les repères au sol des distances au mur. Ces valeurs formeront l'axe des abscisses dans le graphe d'étalonnage.
3. Le téléphone est posé port USB sur le repère, en face de l'extincteur. L'application MARSLogger enregistre pendant 10 s avec les mêmes paramètres sténopés figés pour toutes les séquences d'acquisition.
4. Toutes les acquisitions sont passées à YOLO pour produire les détections sur l'extincteur. Seuls les vrais positifs sont conservés en aval de cette détection.



FIGURE C.4 – Disposition du télémètre laser sur le repère au sol face à l'extincteur.

On peut alors calculer la moyenne et l'écart-type sur toutes les détections pour différents seuils de confiance : 0.25, 0.5, 0.75 comme dans la figure C.5. L'axe des ordonnées indique la proportion en lignes de l'image occupée par une détection : $du = \frac{du_{det}[px]}{total\ lignes[px]}$.

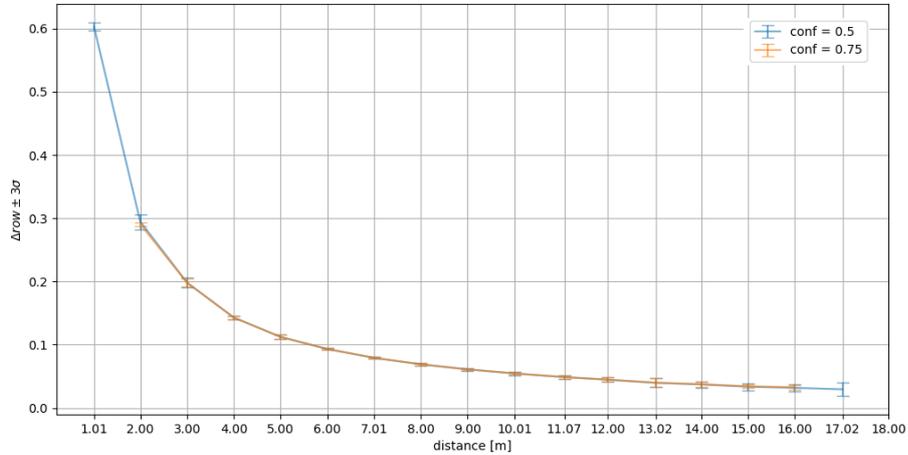


FIGURE C.5 – Fonction d'étalonnage : $du = g(distance)$.

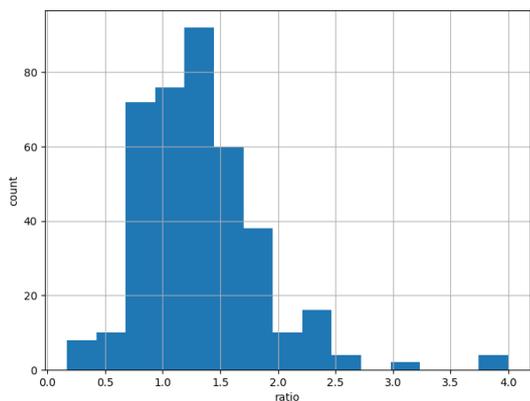
On obtient alors la fonction g tel que $f = g^{-1}$. Cette fonction est obligatoirement monotone.

C.2 Validation par ratio géométrique de la détection

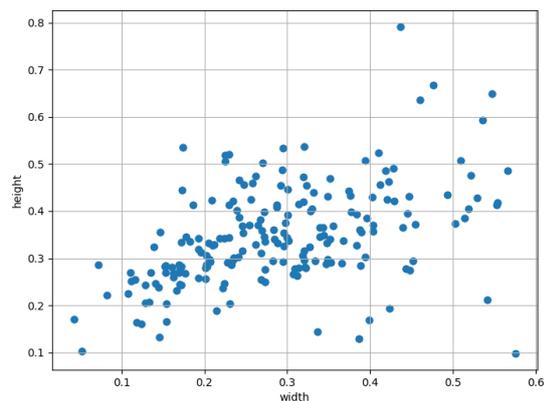
En sachant que les extincteurs sont toujours posés verticalement contre le mur, et qu'ils ont un ratio hauteur/largeur constant, on peut calculer à partir du jeu de données d'entraînement le ratio moyen et son écart-type. On affiche sur les figures C.6a et C.6b les statistiques extraites du jeu de données. L'histogramme permet d'approximer la distribution du ratio des détections par une gaussienne centrée autour de 1.34 et d'écart-type 0.54.

Durant la localisation, le PF utilise les largeurs et hauteurs prédites provenant des sections 8.2.1.1 et 8.2.1.2 pour calculer le ratio prédit. Ce ratio résultera en un poids, $\tilde{\lambda}_t^i$, pour cette validation pour cette particule. Ceci donnerait un nouvel indicateur permettant de valider si la détection est cohérente pour les particules.

Cette validation a été mise en place et testée. Ses résultats ne sont pas satisfaisants à cause des prises d'image qui présentent un extincteur sur le côté ce qui fausse le ratio hauteur/largeur. Cette technique devra être réadaptée pour être validée et intégrée à nos traitements.



(a) Histogramme du ratio hauteur/largeur.



(b) Tracé des hauteurs contre largeurs.

FIGURE C.6 – Statistiques des hauteur/largeur de détections du jeu d'entraînement.

C.3 Apprentissage et YOLO

La figure C.7 présente l'architecture détaillée de YOLO en blocs élémentaires.

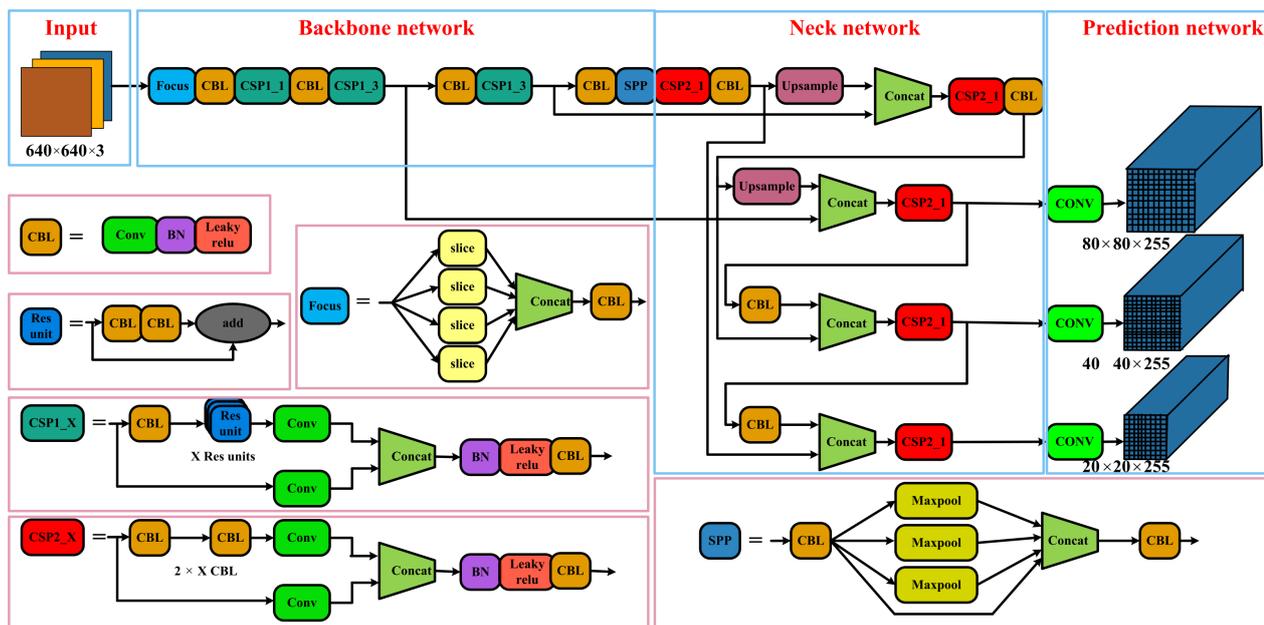


FIGURE C.7 – Architecture de YOLOv5 en sous-blocs élémentaires [Zhang et al., 2023].

- **CBL** : est composé de trois bloc : une convolution 2D, suivi de l'ajout d'un biais sur la valeur de la convolution et le la fonction d'activation nommée "leaky relu".
- **Res unit** : est composé de 2 blocs **CBL** en série avec l'ajout d'un résidu de l'entrée à la sortie de ces blocs.
- **CSP1_X, CSP2_X** : est composé en parties où la première un enchaînement de **CBL**, de groupe de X **Res units** suivis d'une convolution. La deuxième partie est une

- convolution seule. Ces deux parties sont concaténées pour que le résultat passe par un biais, une "leaky relu" et un **CBL**. **CSP2_X** possède 2 fois plus de blocs **CBL**.
- **Focus** : est composé de 4 blocs "slice" qui permet de découper la matrice en entrée et ensuite de la réorganiser via une concaténation. Le tout est suivi par un bloc **CBL**.
 - **SPP** est composé d'un bloc **CBL** suivi de 3 maxpool en parallèle qui sont joint par une concaténation avec un résidu du **CBL** pour ensuite passer dans un **CBL**.

C.4 Décomptes des classes dans COCO

catégorie	classe	décompte	catégorie	classe	décompte
personne	personne	185316	cuisine	verre à vin	5618
véhicule	vélo	4955	cuisine	tasse	14513
véhicule	voiture	30785	cuisine	fourchette	3918
véhicule	moto	6021	cuisine	couteau	5536
véhicule	avion	3833	cuisine	cuillère	4287
véhicule	bus	4327	cuisine	bol	10064
véhicule	train	3159	nourriture	banane	6912
véhicule	camion	7050	nourriture	pomme	4308
véhicule	bateau	7590	nourriture	sandwich	3089
extérieur	feu	9159	nourriture	orange	4597
extérieur	bouche d'incendie	1316	nourriture	brocoli	4927
extérieur	panneau d'arrêt	1372	nourriture	carotte	5539
extérieur	parcètre	833	nourriture	hot-dog	2023
extérieur	banc	6751	nourriture	pizza	4001
animal	oiseau	7290	nourriture	beignet	4977
animal	chat	3301	nourriture	gâteau	4551
animal	chien	3774	meubles	chaise	27147
animal	cheval	4666	meubles	canapé	4113
animal	mouton	6654	meubles	plante en pot	5918
animal	vache	5686	meubles	lit	2905
animal	éléphant	3905	meubles	table	11167
animal	ours	903	meubles	toilette	2873
animal	zèbre	3685	électronique	tv	4036
animal	girafe	3596	électronique	ordinateur portable	3415
accessoire	sac à dos	6200	électronique	souris	1517
accessoire	parapluie	7865	électronique	télécommande	4122
accessoire	sac à main	8778	électronique	clavier	1980
accessoire	cravate	4497	électronique	téléphone portable	4460
accessoire	valise	4251	électronique	micro-ondes	1189
sports	frisbee	1862	électronique	four	2302
sports	skis	4698	électronique	grille-pain	156
sports	snowboard	1960	électronique	évier	3933
sports	ballon de sport	4392	électronique	réfrigérateur	1875
sports	cerf-volant	6560	intérieur	livre	17315
sports	batte de baseball	2400	intérieur	horloge	4328
sports	gant de baseball	2689	intérieur	vase	4623
sports	planche à roulettes	4012	intérieur	ciseaux	1073
sports	planche de surf	4161	intérieur	ours en peluche	3442
sports	raquette de tennis	3411	intérieur	sèche-cheveux	135
cuisine	bouteille	16983	intérieur	brosse à dents	1377

TABLE C.1 – Lists de catégories et leurs classes dans COCO *train2017*.