THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale Information, Structures, Systèmes

Unité de recherche LIRMM(Laboratoire d'Informatique, de Robotique et de Microéléctronique de Montpellier)

Cartes topologiques sémantiques pour la navigation visuelle en robotique autonome

Présentée par Fama NGOM

le 12 Décembre 2025

Sous la direction de Marianne HUCHARD et Karen GODARY-DEJEAN

Devant le jury composé de

Luc FABRESSE, Professeur, IMT Nord Europe	Rapporteur
Luc JAULIN, Professeur, Université Bretagne Occidentale	Rapporteur
Jacques MALENFANT, Professeur, Sorbonne Université	Examinateur
Frédéric COLLEDANI, PhD Ingénieur de recherche, CEA Centre de Saclay	Examinateur
Abdelhak-Djamel SERIAI, Professeur, Université de Montpellier	Examinateur
Lei ZHANG, PhD, CEO euroDAO	Co-encadrant
Yulin (Huaxi) ZHANG, Maîtresse de conférences, Université de Picardie Jules Verne	Invitée

UNIVERSITÉ DE MONTPELLIER

Résumé

Cette thèse s'intéresse à la navigation sémantique explicable pour les robots mobiles évoluant dans des environnements intérieurs dynamiques, caractérisés par des changements imprévisibles tels que le déplacement d'objets ou le blocage de chemins. Elle propose IntelliMap, un framework de cartographie hiérarchique destiné à générer automatiquement des cartes HSTM (Hierarchical Semantic Topometric Maps) à partir de données brutes. Ces cartes intègrent des informations métriques, topologiques et sémantiques et peuvent être adaptées en fonction des objectifs de navigation.

À partir de cette représentation hiérarchique, le framework IntelliNav proposé assure une navigation sémantique adaptative et explicable. Il repose sur une planification globale construite à partir d'éléments sémantiques, permettant de définir un chemin composé de plusieurs sous-objectifs successifs jusqu'à l'objectif final. Chaque sous-objectif est ensuite atteint par une navigation locale s'appuyant sur la couche métrique de la HSTM. Pendant l'exécution, le système intègre un mécanisme de replanification dynamique en cas d'obstacles persistants, une mise à jour continue de la carte HSTM, ainsi qu'une mémoire persistante exploitant l'historique des missions passées.

L'ensemble a été évalué sur des scènes simulées réalistes (HM3DSem, uHumans2), comprenant des environnements multi-pièces et multi-étagés, dans des conditions dynamiques. Les résultats expérimentaux mettent en évidence des gains significatifs en termes de taux de réussite, d'adaptabilité et de lisibilité du comportement du robot, par rapport à l'état de l'art.

Enfin, une extension distribuée, nommée MultiNav, est introduite en tant que perspective théorique en annexe. Elle établit les fondations d'une architecture multi-robot reposant sur des cartes locales synchronisées, ouvrant la voie à un déploiement collaboratif à grande échelle dans des environnements industriels, hospitaliers ou de bureaux.

Mots-clés : navigation sémantique, cartographie hiérarchique, mise à jour de carte, replanification dynamique, explicabilité, robotique mobile, environnement dynamique, système multi-robots.

Abstract

This thesis focuses on explainable semantic robot navigation for mobile robots operating in dynamic indoor environments, where objects or paths may change unpredictably. It introduces IntelliMap, a hierarchical mapping framework designed to generate Hierarchical Semantic Topometric Maps (HSTM) from raw data. This hierarchical map merges metric, topological, and semantic information, while allowing customization of the semantic level according to navigation objectives. The framework supports complex, potentially multi-floor environments and enables dynamic updates to the map structure.

Based on this hierarchical representation, the proposed IntelliNav framework enables adaptive and explainable semantic robot navigation. It relies on a semantic global planning process constructed from semantic elements, defining a path composed of multiple successive subgoals leading to the final target. Each subgoal is then reached through local navigation based on the metric layer of the HSTM. During execution, the system integrates a dynamic replanning mechanism in response to persistent obstacles, continuous updates to the HSTM, and persistent memory that leverages past mission history.

The entire system has been evaluated in realistic simulated environments (HM3DSem, uHumans2), involving multi-room and multi-floor configurations under dynamic conditions. Compared to the state of the art, experimental results demonstrate significant improvements in success rate, adaptability, and the interpretability of the robot's behavior.

Finally, a distributed extension named MultiNav is introduced as a theoretical perspective in the appendix. It lays the foundation for a multi-robot architecture based on synchronized local maps, paving the way for large-scale collaborative deployment in industrial, hospital, or office environments.

Remerciements

J'exprime tout d'abord ma profonde gratitude à mes directrices de thèse, Madame Marianne Huchard et Madame Karen Godary-Dejean, pour leur accompagnement tout au long de ce travail, qui a été essentiel à l'aboutissement de cette recherche. Je les remercie sincèrement pour leurs relectures attentives et méticuleuses, leur bienveillance constante et l'intérêt qu'elles ont toujours porté à l'avancée de mes travaux. Leurs remarques et suggestions ont largement contribué à améliorer la qualité de ce manuscrit.

Je souhaite également remercier ma co-encadrante, Madame Huaxi Yulin Zhang, pour la qualité de ses échanges et pour m'avoir initié au domaine de la robotique mobile. Le partage de ses connaissances et de son expertise dans ce domaine a été particulièrement formateur.

Je souhaite aussi remercier Monsieur Lei Zhang pour la confiance qui m'a été accordée pour ce sujet de thèse CIFRE avec l'entreprise euroDAO. Cette confiance m'a permis de mener mes travaux dans un contexte industriel stimulant.

Je tiens à remercier sincèrement l'ensemble des membres du jury pour avoir accepté d'évaluer ce travail et pour la qualité de leurs échanges lors de la soutenance. J'adresse en particulier mes remerciements aux rapporteurs, Monsieur Luc Jaulin et Monsieur Luc Fabresse, pour le temps consacré à la relecture attentive de ce manuscrit et pour leurs commentaires constructifs. Je remercie également Monsieur Jacques Malenfant pour sa participation à mes comités de suivi individuel (CSI) et pour les échanges constructifs ainsi que l'ensemble des membres du CSI.

Mes remerciements vont également à ma famille et à mes amis pour leur soutien tout au long de ce parcours. Je remercie tout particulièrement celles et ceux qui m'ont permis de m'évader du travail de recherche à travers des moments de partage et de convivialité, et qui m'ont encouragé à avancer malgré les exigences de ce parcours.

Merci à Monsieur et Madame Mbaye pour leurs conseils, leur soutien et leur bienveillance.

Je remercie mon amie, Maella , pour sa présence constante, les nombreuses heures passées au téléphone, ses conseils précieux et son soutien face aux difficultés rencontrées au cours de cette thèse.

Je souhaite adresser une pensée toute particulière à mes parents, pour leurs prières, leur patience et leur soutien indéfectible. Je remercie tout spécialement Badiane, à qui je dédie ce travail, pour avoir toujours insisté sur l'importance de mes études et pour s'être assuré que je puisse poursuivre mon parcours académique dans les meilleures conditions. Mes remerciements vont également à Kader et Youssou pour leur soutien constant et leurs encouragements dans la poursuite de mes études. Je remercie Oussou, mon répétiteur, qui m'a initié aux études avant mon entrée à l'école et a joué un rôle déterminant dans mon parcours scolaire. Je remercie également Badiane Aminta, qui m'a accompagné durant mes premières années à l'école du Manguier, pour son soutien précieux à cette étape fondatrice. J'adresse une pensée émue à feu Sadiaraw Diouf et feu Ablaye Diouf, qui m'ont toujours encouragé depuis mon plus jeune âge. J'aurais profondément souhaité que vous soyez présents pour assister à l'aboutissement de ce travail.

Je remercie Monsieur Babacar Gueye pour son accueil chaleureux lors de mes premiers jours en France, ainsi que l'ensemble de sa famille pour leur bienveillance et leur soutien.

Enfin, je remercie chaleureusement mon cher Monsieur Gueye pour son soutien quotidien indéfectible et son enthousiasme contagieux, tant à l'égard de mes travaux que de la vie en général. Ses conseils avisés, sa disponibilité constante, sa compréhension et son soutien, dans les moments heureux comme dans les périodes plus difficiles, ont été d'un grand réconfort tout au long de ce parcours. Son accompagnement de tous les jours a été une source précieuse de motivation et de force.

Sommaire

1 Introduction **17**

Introduction **17**

1.1 Problématique et contribution 19
1.2 Structure de la thèse 20
1.3 Contributions et publications associées 22

2 Cartographie sémantique avec IntelliMap **24**

2.1 Introduction 24
2.2 État de l'art 25
2.2.1 les principaux types de cartes 25
2.2.1.1 Cartes métriques 25
2.2.1.2 Cartes topologiques 29
2.2.2 Approches modernes de cartographie sémantique 30
2.2.2.1 Cartes topométriques hiérarchiques 30
2.2.2.2 Graphe de scène 3D (3D Scene Graph) 31
2.2.3 Limites des approches existantes 32
2.2.4 Synthèse comparative : couverture de l'information sémantique 33
2.3 Framework IntelliMap : architecture 34
2.3.1 Structure de la HSTM 35
2.3.1.1 Couche métrique 36
2.3.1.2 Couche des objets 36
2.3.1.3 Couche des pièces et des bâtiments 37
2.4 Processus de construction d'une carte avec IntelliMap 42
2.4.1 Données des capteurs : acquisition et intégration 43
2.4.2 SLAM Métrique : génération de la Carte Métrique 43
2.4.3 Cartographie des objets 45

2.4.3.1 Détection d'objets . . . 45
2.4.3.2 Vérification de l'unicité des objets . . . 46
2.4.3.3 Processus de cartographie des objets . . . 48
2.4.4 Organisation spatiale et cartographie relationnelle . . . 48
2.4.4.1 Segmentation des pièces . . . 48
2.4.4.2 Catégorisation des lieux . . . 49
2.4.4.3 Processus de génération des arêtes . . . 51
2.4.4.4 Segmentation des bâtiments . . . 52
2.5 Conclusion . . . 53

3 Navigation Sémantique avec IntelliNav 56
3.1 Introduction . . . 56
3.2 État de l'art . . . 57
3.2.1 Planification statique vs planification adaptative . . . 58
3.2.1.1 Planification statique . . . 58
3.2.1.2 Planification adaptative . . . 60
3.2.2 Navigation guidée par le langage naturel et les LLM . . . 61
3.2.3 Limites des approches traditionnelles en navigation sémantique . . . 62
3.2.4 Navigation sémantique dynamique . . . 63
3.2.5 Stratégies de replanification dans la navigation sémantique . . . 65
3.2.6 Limites des approches existantes . . . 66
3.3 Framework IntelliNav : structure et fonctionnalités . . . 67
3.3.1 Missions et états du robot . . . 68
3.3.1.1 Les differents états définis dans IntelliNav . . . 69
3.3.1.2 Transitions entre états . . . 70
3.3.2 Localisation sémantique . . . 71
3.3.2.1 Introduction . . . 71
3.3.2.2 Processus de localisation avec IntelliNav . . . 72
3.3.3 Planification globale de chemin sémantique . . . 74
3.3.3.1 Scénario 1 : navigation vers un objet ou lieu déjà cartographié . . . 76
3.3.3.2 Scénario 2 : exploration pour des objets non cartographiés . . . 76
3.3.3.3 Scénario 3 : exploration multi-cibles . . . 77
3.3.4 Planification des sous-objectifs et navigation locale . . . 77
3.3.4.1 Navigation locale basée sur une costmap . . . 78
3.3.4.2 Validation d'un sous-objectif . . . 79
3.3.4.3 Cas particulier : navigation vers un objet . . . 79
3.3.4.4 Navigation locale lors des transitions inter-étages . . . 79
3.3.5 Mise à jour de la carte sémantique HSTM . . . 80
3.3.5.1 Mise à jour des états des arêtes . . . 81
3.3.5.2 Mise à jour des attributs des nœuds . . . 82

3.3.5.3 Gestion des blocages successifs . . . 84
3.3.6 Replanification du chemin global sémantique . . . 85
3.3.6.1 Blocage de chemin : définition . . . 85
3.3.6.2 Processus de replanification . . . 87
3.4 Conclusion . . . 88

4 Expérimentation et évaluation 89
4.1 Protocole expérimental . . . 89
4.2 IntelliMap : résultats des expérimentations . . . 90
4.2.1 Précision de la cartogaphie . . . 90
4.2.2 Analyse de l'empreinte mémoire du stockage de la HSTM . . . 92
4.2.2.1 Résultats sur le dataset "uHumans2 Office" . . . 92
4.2.2.2 Résultats sur le dataset HM3DSem . . . 93
4.3 IntelliNav : résultats des expérimentations . . . 95
4.3.1 Optimisation de la navigation par validation sémantique des pièces . . . 95
4.3.2 Exploration et découverte d'objets non cartographiés . . . 95
4.3.3 Performance computationnelle de la planification sémantique globale des trajets . . . 96
4.3.3.1 Évaluation de la planification vers des objectifs cartographiés . 97
4.3.3.2 Évaluation de la planification vers des objets non cartographiés 97
4.3.4 Optimisation dynamique des trajectoires par ajustement des états des connexions . . . 98
4.3.4.1 Expérimentation sur un environnement à un étage . . . 98
4.3.4.2 Expérimentation sur un environnement multi-étages . . . 101
4.3.5 Gestion des objets manquants ou dynamiques . . . 103
4.4 Conclusion . . . 106

5 Conclusion, perspectives et valorisation industrielle 107
5.1 Synthèse des contributions . . . 107
5.2 Impact industriel et pertinence des choix architecturaux pour l'entreprise euroDAO 108
5.2.1 Exemple d'intégration dans un hôpital multi-étages . . . 110
5.2.2 Exemple d'intégration dans un site industriel . . . 111
5.3 Perspectives . . . 112

Bibliographie 114

A MultiNav, navigation multi-robot distribuée basée sur la HSTM 123
A.1 Introduction . . . 123
A.2 Etat de l'art . . . 124
A.2.1 Reconstruction dense et cartographie collaborative . . . 124
A.2.2 Exploration multi-robot d'environnements inconnus . . . 125

A.2.3 Navigation et planification multi-robot avec raisonnement sémantique . . 126
A.2.4 Limite des approches et proposition 126
A.3 Architecture 126
A.3.1 Adaptation de la structure de la HSTM au contexte multi-robot 127
A.3.2 Carte HSTM locale 129
A.3.3 Module de navigation IntelliNav 130
A.3.4 Module de décision de synchronisation locale (Sync Decision) 130
A.3.5 Module de Synchronisation conditionnelle 131
A.3.5.1 Traitement des transitions et envoi vers la carte HSTM globale : 132
A.3.5.2 Téléchargement des mises à jour vers les cartes locales : 134
A.3.5.3 Exemple concret de mise à jour et gestion temporelle des conflits :134
A.3.6 HRI : interaction humain robot 135
A.4 Conclusion 135

Liste des figures

1 IntelliMap génère une carte hiérarchique sémantique (HSTM) à partir des données environnementales brutes. Cette carte sert d'entrée à IntelliNav, qui réalise une navigation sémantique avec mise à jour dynamique appliquée directement sur la HSTM pour refléter les changements détectés dans l'environnement. . . . 19

2 Illustration du processus de navigation sémantique dynamique de IntelliNav, basé sur la HSTM obtenue avec IntelliMap. 21

3 Exemple de carte de grille d'occupation 2D générée à partir de l'étage 0 de la scène 00862 du dataset HM3DSem. 26

4 Nuage de points 3D acquis dans un environnement industriel à l'aide du robot Scout équipé d'un Light Detection and Ranging (LiDAR) Hesai Pandar XT-32. Cette capture est réalisée dans une usine réelle. 27

5 Carte de maillage du premier étage des anciens bureaux de l'entreprise euroDAO. 28

6 Illustration du concept de carte topologique pour la navigation robotique. Chaque nœud représente un lieu caractérisé par des informations sensorielles (ex. image panoramique) collectées lors du premier passage du robot. Image obtenue depuis le site du laboratoire de l'Université de Bielefeld `https://www.ti.uni-bielefeld.de/html/research/topmaps/index.html`. 30

7 À partir d'un nuage de points 3D en entrée (1), différentes représentations hiérarchiques sont générées : une carte topologique 0D (2), une carte 1D avec sommets annotés par leurs coordonnées (3), une carte 2D segmentée en régions colorées (4) et une carte 3D contenant des informations volumétriques (5). Image obtenue de [1]. 31

8 Schéma de l'architecture du framework IntelliMap. Ce schéma illustre les principaux modules du framework IntelliMap, conçu pour générer une carte HSTM à partir de données capteurs brutes. Le processus comprend la segmentation des pièces et des bâtiments, la catégorisation des lieux, la cartographie des objets et la SLAM métrique. Les résultats intermédiaires, carte topologique sémantique et carte métrique sémantique, sont fusionnés pour produire une représentation hiérarchique intégrant les aspects topologiques, sémantiques et métriques de l'environnement. 35
9 Visualisations sémantiques des objets détectés dans l'environnement d'un bureau extrait du dataset UHumans2. 45
10 Exemple de détection d'objet avec Omni3D dans le couloir de l'environnement Office du dataset UHuman2. L'objet détecté est une chaise (chair), encadrée par une boîte englobante rose. 46
11 Exemple de catégorisation d'une pièce avec GPT-4o 51
12 Cartographie des pièces 52

13 IntelliNav Framework 67
14 Couche de la HSTM utilisé par IntelliNav 68
15 IntelliNav : planification sémantique globale 68
16 IntelliNav : description des mises à jour de la HSTM 69
17 IntelliNav : Navigation locale 69
18 Les differents états du robot lors d'une mission 69
19 Exemple de localisation sémantique : Correspondance entre la position du robot et la pièce associée. 72
20 Détection d'objet 73
21 Planification globale du chemin 74
22 Exemple de navigation locale entre deux pièces sur une costmap 78
23 Replanification du chemin global sémantique 86
24 Exemple de visualisation d'un obstacle sur une porte dans une scene du dataset HM3DSem. Le ballon rouge représente l'obstacle. 87

25 Illustration de l'évaluation de la précision du mapping avec IntelliMap à partir de détections fournies par Omni3D. 91
26 Configuration par défaut : Trajet sémantique global de référence, de living_room1 à microwave, calculé avec les poids par défaut des arêtes de la Hierarchical Semantic Topometric Map (HSTM) de la scène 00824. 99
27 Comparaison des trajets sémantiques globaux avec des poids de nœuds variables. Les sous-figures illustrent l'impact de différentes configurations de poids sur le trajet sémantique global. 100
28 Liste des pièces de la scène 00843 de la troisième couche de la HSTM 102
29 Liste des pièces de la scène 00862 de la troisième couche de la HSTM 103

30 Modification du maillage de la scène 00824 pour simuler la disparition dynamique de l'objet *toilet* dans *bathroom1*. 104
31 L'objet *toilet* a été détecté avec succès dans *bathroom2*, confirmant la réussite de la mission. 105

32 Architecture distribuée proposée pour la navigation multi-robot basée sur la HSTM. 128
33 Processus de décision du module de synchronisation pour la mise à jour conditionnelle de la HSTM globale 133

Liste des tableaux

2.1 Analyse comparative des configurations de couches sur IntelliMap et d'autres méthodes existantes. 34
2.2 Résultats de catégorisation des lieux : comparaison entre les sorties ChatGPT et les types de pièces de référence. 55

3.1 Association des états des arêtes et des poids 82
3.2 Transitions d'états des arêtes dans IntelliNav 83

4.1 Comparaison de l'utilisation mémoire (MU) avec et sans carte métrique. IntelliMap utilise une costmap comme carte métrique, tandis qu'Hydra utilise un maillage (mesh). 93
4.2 Comparaison de l'empreinte mémoire pour le stockage des cartes. 94
4.3 Comparaison de la longueur des trajectoires et de la réduction 95
4.4 Impact des modifications de l'état des bords sur la planification sémantique des chemins dans les scénarios à un étage et à plusieurs étages. 98

A.1 Transitions locales déclenchant une synchronisation vers la HSTM globale 131

Acronymes

HSTM Hierarchical Semantic Topometric Map. 12, 13, 19, 20, 25, 34–36, 42, 49, 52, 53, 56, 57, 60, 67, 68, 70–82, 84, 85, 87, 88, 92–97, 99, 104–106, 108, 109, 124, 126, 127, 129–136

LiDAR Light Detection and Ranging. 11, 27, 28, 43, 44, 53, 60, 90

LLM Large Language Model. 50, 61, 62, 64, 66, 76, 77, 105, 124

SLAM Simultaneous Localization and Mapping. 25, 36, 43, 44, 53, 65, 124–126

CHAPITRE 1

Introduction

Les travaux de recherche de cette thèse ont été menés dans le cadre du dispositif CIFRE, en collaboration étroite avec la société euroDAO SAS, entreprise deeptech fondée en France en 2020, et actuellement implantée à Montpellier. euroDAO conçoit des systèmes autonomes intelligents combinant navigation robotique, contrôle partagé et intelligence artificielle distribuée, reposant notamment sur des architectures Edge AI et Cloud AI. Ses solutions sont destinées à des applications avancées dans les domaines industriel, agricole et spécial, où la robustesse, la sécurité et l'adaptabilité sont des exigences critiques.

Cette collaboration a offert un cadre privilégié de co-innovation entre recherche académique et besoins industriels. Elle a permis d'ancrer les contributions scientifiques de la thèse dans des cas d'usage concrets, tout en bénéficiant des retours terrain pour ajuster les hypothèses de conception. L'implication continue d'euroDAO dans toutes les phases du projet — de la définition des problématiques à la valorisation des résultats — a favorisé une dynamique de co-ingénierie, renforçant à la fois la pertinence académique et la transférabilité industrielle des solutions proposées.

La problématique générale est l'assistance des individus dans des espaces domestiques ou la conduite d'opérations autonomes dans des environnements industriels, tâches pour lesquelles la navigation autonome des robots constitue un défi technologique majeur. Les robots doivent être capables de percevoir et d'interpréter leur environnement afin de se déplacer de manière efficace et sûre tout en s'adaptant aux changements dynamiques et imprévus qui s'y produisent.

Traditionnellement, les systèmes de navigation reposent sur des représentations géométriques de l'environnement, où les robots perçoivent leur espace en termes de distances, d'angles et de formes. Bien que ces approches aient permis des avancées significatives, elles peuvent montrer leurs limites dans des environnements réels, où la complexité et la variabilité des scènes nécessitent une compréhension plus riche et plus intuitive.

Dans ce contexte, la navigation sémantique, qui associe des informations sémantiques à des

représentations environnementales, permet aux robots de dépasser la simple perception géométrique pour accéder à une compréhension conceptuelle de leur environnement. Par exemple, un robot équipé de capacités sémantiques ne se contente pas de détecter une table comme un ensemble de points dans l'espace ; il comprend qu'il s'agit d'un meuble utilisé pour poser des objets, et qu'il est souvent situé dans une cuisine ou un salon. Cette compréhension enrichie ouvre la voie à des décisions de navigation plus intelligentes et plus adaptatives, où le robot peut raisonner sur les objets, leur utilité et leurs relations. Ainsi, la navigation sémantique offre une opportunité d'accroître l'intelligence et l'autonomie des robots, en facilitant une interaction plus naturelle et plus efficace avec leur environnement.

Cependant, les environnements intérieurs dynamiques présentent des défis importants pour la navigation sémantique. Les espaces réels sont caractérisés par une variabilité constante et des changements imprévisibles. Ces dynamiques peuvent prendre plusieurs formes, chacune posant des défis spécifiques pour les robots mobiles. Par exemple, dans un environnement intérieur, les portes peuvent être fermées ou ouvertes à tout moment, modifiant ainsi l'accessibilité des pièces. Des objets peuvent être déplacés ou laissés dans des endroits inattendus, bloquant temporairement un couloir, une porte ou un passage. Certains objets dans un environnement intérieur ne sont pas statiques. Ils peuvent être déplacés, ajoutés ou retirés, modifiant ainsi la configuration spatiale. Certains de ces obstacles ne sont pas permanents, mais ils nécessitent une adaptation immédiate de la part du robot pour éviter les collisions ou les échecs de mission. Ces dynamiques rendent les environnements intérieurs particulièrement difficiles à naviguer. Les robots doivent non seulement percevoir et interpréter ces changements, mais aussi ajuster leur stratégie de navigation en conséquence.

Ces défis soulignent la nécessité d'approches innovantes, pour permettre aux robots de fonctionner efficacement dans des environnements réels et complexes.

Un autre défi majeur dans le domaine de la robotique mobile est l'explicabilité des choix de navigation. Dans ce contexte, l'explicabilité désigne la capacité du système à justifier ses décisions de manière compréhensible pour un utilisateur humain. Elle englobe notamment la traçabilité des actions, l'interprétabilité, c'est-à-dire le degré auquel un humain peut comprendre les causes d'une décision et, plus largement, la transparence des mécanismes décisionnels. Dans de nombreux contextes, notamment les applications domestiques et industrielles, cette aptitude à rendre les choix du robot intelligibles est essentielle. Elle contribue à renforcer la confiance dans les systèmes autonomes et à faciliter leur adoption dans des environnements réels.

Dans un cadre domestique, les robots interagissent directement avec les utilisateurs, souvent sans supervision technique. Par exemple, un robot de service doit être capable d'expliquer pourquoi il a choisi un chemin spécifique pour atteindre une pièce, ou pourquoi il a évité un certain obstacle. Ces explications permettent aux utilisateurs de comprendre les actions du robot et de s'assurer qu'il fonctionne de manière sûre et efficace. Sans cette transparence, les utilisateurs pourraient perdre confiance dans le système, ce qui limiterait son adoption.

Dans les environnements industriels, l'explicabilité des choix de navigation est tout aussi importante, voire plus critique. Les robots sont souvent déployés dans des contextes où la sécurité

et l'efficacité sont primordiales, comme les entrepôts logistiques ou les usines de production. Dans ces environnements, les décisions de navigation doivent non seulement être optimales, mais aussi transparentes et justifiables. L'explicabilité des choix de navigation joue un rôle clé dans le renforcement de la confiance entre les robots et les utilisateurs. De plus, l'explicabilité facilite l'interaction humain-robot, car elle aide les utilisateurs à comprendre et à anticiper les actions du robot.

Par ailleurs, le développement de solutions de navigation efficaces doit également tenir compte des contraintes computationnelles imposées par les plateformes robotiques embarquées. Dans de nombreux cas, les robots opèrent sans connexion continue à un serveur distant ou à une infrastructure cloud, ce qui impose de concevoir des systèmes capables de fonctionner de manière autonome et locale. Cela nécessite l'utilisation de représentations compactes de l'environnement, optimisées pour un usage mémoire limité et capables de prendre en charge des mises à jour dynamiques et réactives en cas de changements dans l'environnement.

Un système de navigation capable de gérer ces contraintes tout en conservant une bonne capacité d'adaptation est essentiel pour garantir une autonomie robuste, évolutive et déployable dans des conditions réelles.

1.1 Problématique et contribution

La navigation robotique dans des environnements dynamiques nécessite de repenser la manière dont les cartes sont conçues, mises à jour et exploitées pour la planification des trajectoires. Cette thèse s'inscrit dans ce cadre et vise à répondre à la question suivante : "Comment concevoir une navigation sémantique explicable pour un robot mobile opérant dans des environnements intérieurs dynamiques, où des objets ou des chemins peuvent changer de manière imprévisible ?"

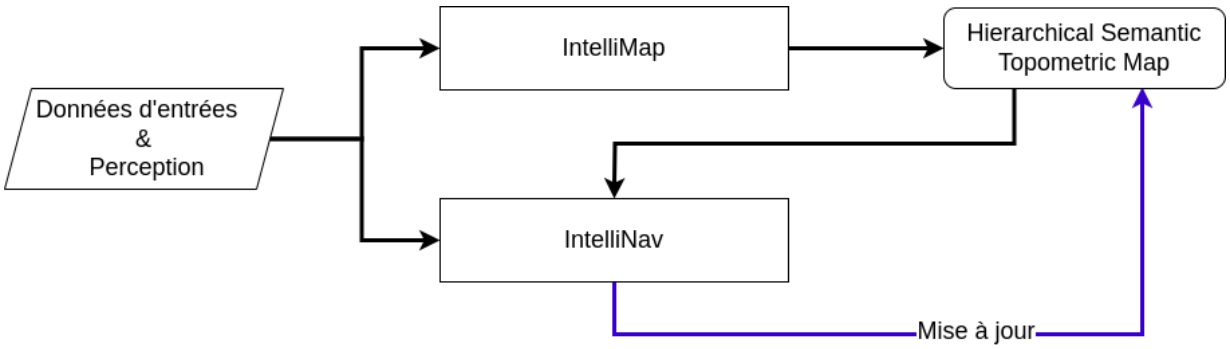


FIGURE 1 – IntelliMap génère une carte hiérarchique sémantique (HSTM) à partir des données environnementales brutes. Cette carte sert d'entrée à IntelliNav, qui réalise une navigation sémantique avec mise à jour dynamique appliquée directement sur la HSTM pour refléter les changements détectés dans l'environnement.

Pour répondre à cette problématique, cette thèse propose deux méthodes principales :

— Développement d'un framework de cartographie sémantique : IntelliMap est un framework hiérarchique conçu pour générer des cartes enrichies sémantiquement (HSTM)

intégrant des informations métriques et sémantiques. Cette carte, structurée en plusieurs niveaux (bâtiments, pièces, objets, métrique), vise à répondre aux besoins variés de la navigation tout en garantissant une gestion efficace des ressources computationnelles et une mise à jour dynamique.

— Conception d'un framework de navigation sémantique : IntelliNav exploite les informations issues de IntelliMap pour permettre une navigation adaptative et explicable. Ce framework intègre des mécanismes de replanification dynamique, une mise à jour en temps réel des cartes et une mémoire persistante. Cette dernière permet au robot de tirer parti des expériences passées pour améliorer ses performances futures.

La figure 1 illustre cette relation entre les deux frameworks : IntelliMap produit une représentation hiérarchique HSTM de l'environnement, qui sert d'entrée à IntelliNav pour guider la navigation sémantique adaptative.

La figure 2 illustre plus en détail le fonctionnement du système proposé à travers un scénario de navigation sémantique. À partir d'une instruction du type « Depuis le salon, va aux toilettes », le robot génère un chemin global sémantique composé de pièces connectées. Si un blocage est détecté durant l'exécution du chemin (par exemple entre living_room et bedroom), un processus de replanification dynamique est déclenché pour déterminer un chemin alternatif via d'autres pièces. La navigation locale vers chaque sous-objectif est réalisée à l'aide de la costmap (ou carte des coûts, grille numérique de l'environnement où chaque cellule contient un coût de traversée), et la HSTM est continuellement mise à jour en fonction des observations visuelles et spatiales. L'ensemble du système repose sur cette interaction fluide entre représentation hiérarchique, perception, planification adaptative et exécution.

Les hypothèses principales de cette thèse sont les suivantes :

— Une représentation hiérarchique intégrant des informations métriques et sémantiques pourrait améliorer la précision de la navigation, faciliter les processus de planification et optimiser la gestion des mises à jour dans des environnements dynamiques.

— La combinaison d'une replanification dynamique, d'une mémoire persistante et d'une navigation par sous-objectifs renforce l'adaptabilité et l'explicabilité de la navigation sémantique dans des scénarios réels.

1.2 Structure de la thèse

Ce manuscrit s'articule autour de cinq chapitres principaux, correspondant aux grandes étapes du développement de cette recherche.

Le chapitre 2, *Cartographie sémantique avec IntelliMap* explore les fondements de la cartographie sémantique en robotique, tout en mettant en lumière les limites des approches existantes. Il présente en détail le framework IntelliMap, notamment sa structure hiérarchique (HSTM) pour répondre aux besoins variés des scénarios de navigation. Enfin, une validation expérimentale est réalisée à travers différents scénarios pour démontrer l'efficacité et l'adaptabilité du framework.

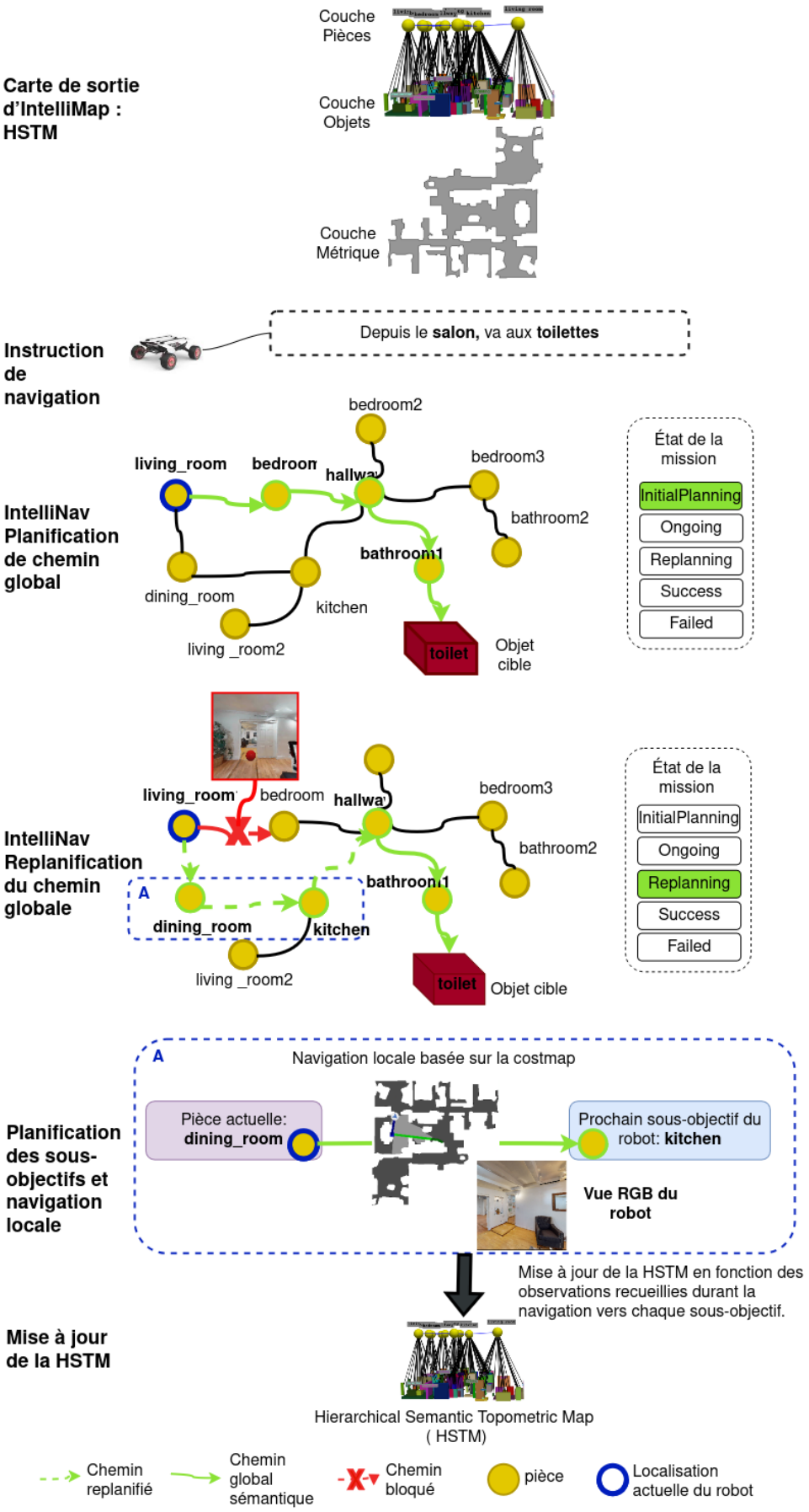


FIGURE 2 – Illustration du processus de navigation sémantique dynamique de IntelliNav, basé sur la HSTM obtenue avec IntelliMap.

Le chapitre 3, *Navigation adaptative avec IntelliNav*, exploite les cartes HSTM générées pour permettre une navigation adaptative et explicable dans des environnements dynamiques. Après une analyse critique des approches de navigation existantes et de leurs limites, le framework IntelliNav est introduit. Ce chapitre décrit ses mécanismes, notamment la navigation globale et locale, la replanification dynamique et la mise à jour en temps réel des cartes. Les performances du framework sont évaluées expérimentalement dans divers scénarios, mettant en évidence sa capacité à s'adapter aux changements imprévus et à améliorer la fiabilité des trajets.

Le chapitre 4, *Expérimentation et évaluation*, traite de la performance des frameworks IntelliMap et IntelliNav. Il présente les protocoles expérimentaux mis en place pour tester l'efficacité des deux modèles dans divers scénarios représentatifs d'environnements. Les résultats obtenus sont analysés à travers plusieurs indicateurs de performance.

Enfin, le chapitre 5, *Conclusion, perspectives et valorisation industrielle*, clôture ce manuscrit en offrant une analyse critique des résultats obtenus, une mise en perspective des contributions, et des orientations pour les travaux futurs. Une extension théorique du système vers un fonctionnement multi-robot distribué, nommée MultiNav, est également introduite dans les perspectives. Les premières spécifications de cette architecture, fondée sur une synchronisation conditionnelle des cartes hiérarchiques et ouvrant la voie à une navigation collaborative à grande échelle, sont présentées en annexe.

1.3 Contributions et publications associées

Les travaux réalisés dans cette thèse ont donné lieu aux contributions suivantes :

— IntelliMap : Un framework de cartographie hiérarchique, validé sur des environnements complexes tels que des bureaux et des sites industriels.

— IntelliNav : Un framework de navigation sémantique adaptatif, intégrant des mécanismes de mise à jour dynamique et de replanification.

Ces contributions ont fait l'objet des rédactions suivantes :

— Article 1 : Publication acceptée à la conférence TAROS, présentant les performances du framework IntelliMap ainsi qu'un algorithme de planification globale sémantique de chemin.

> *Ngom, F., Zhang, H.Y., Zhang, L., Godary-Dejean, K., Huchard, M. (2025). IntelliMove : Enhancing Robotic Planning with Semantic Mapping. In : Huda, M.N., Wang, M., Kalganova, T. (eds) Towards Autonomous Robotic Systems. TAROS 2024. Lecture Notes in Computer Science(), vol 15051. Springer, Cham. https ://doi.org/10.1007/978-3-031-72059-8_7*

— Article 2 : Manuscrit portant sur l'évaluation de la navigation sémantique avec IntelliNav qui sera soumis prochainement à un journal.

> *Ngom, F., Zhang, H.Y., Zhang, L., Godary-Dejean, K., Huchard, M. (2025). IntelliNav : A Dynamic Semantic Robot Navigation Framework Using a Hierarchical*

Semantic Topometric Map for Complex Indoor Environments.

CHAPITRE 2

Cartographie sémantique avec IntelliMap

2.1 Introduction

La cartographie sémantique en robotique vise à dépasser les limites des représentations purement géométriques en intégrant des informations contextuelles et sémantiques sur l'environnement. Contrairement aux cartes métriques traditionnelles, qui ne contiennent que des données spatiales, une carte sémantique associe aux éléments cartographiés des catégories issues de classes connues [2]. Cette structuration permet d'enrichir la représentation de l'environnement en reliant les informations spatiales à des concepts abstraits, tels que les types d'objets, les relations fonctionnelles entre les espaces et les interactions possibles entre les éléments présents. Par exemple, la présence d'une télévision dans une pièce suggère qu'il s'agit probablement d'un salon, tandis qu'un réfrigérateur indique une cuisine. Les entités cartographiées peuvent être associées à une base de connaissances indépendante du contenu de la carte, offrant ainsi des capacités de raisonnement avancées pour l'interprétation et la prise de décision. Cette approche permet aux robots de mieux comprendre leur environnement et d'adapter leur navigation sémantique et leurs interactions en fonction des informations contextuelles disponibles.

La cartographie sémantique constitue une base essentielle pour la planification des trajets et l'interaction homme-robot. Dans notre approche, afin de mieux répondre aux contraintes des environnements dynamiques, nous faisons le choix d'une structuration hiérarchique des cartes, facilitant ainsi les mises à jour en temps réel tout en limitant les coûts computationnels. Cette organisation permet également de s'adapter aux objectifs variés de la navigation sémantique, qu'il s'agisse de relier des objets spécifiques ou de naviguer entre différentes pièces, étages ou bâtiments. Ces besoins motivent le développement du framework IntelliMap, conçu pour intégrer ces capacités au sein d'une cartographie sémantique hiérarchique modulaire.

La carte finale produite par IntelliMap, appelée Hierarchical Semantic Topometric Map

(HSTM), est hautement adaptable. Elle peut intégrer deux ou plusieurs couches en fonction du niveau d'informations sémantiques requis par une application donnée. Cette flexibilité autorise une personnalisation de la carte en fonction des besoins variés, en combinant des données géométriques précises avec une compréhension contextuelle des relations spatiales et des objets.

Par exemple, si le robot doit uniquement naviguer entre différentes pièces, la couche des objets peut être exclue, réduisant ainsi la complexité du modèle tout en conservant une structure topologique suffisante pour la navigation inter-pièces. En revanche, si le robot doit naviguer à travers plusieurs bâtiments distincts, une couche sémantique intégrant les bâtiments devient essentielle pour organiser la carte à un niveau plus abstrait. Cette hiérarchisation permet d'adapter dynamiquement la représentation cartographique en fonction du contexte de navigation et des exigences opérationnelles.

Le présent chapitre est organisé comme suit : La section 2.2 est consacrée à un état de l'art présentant les principaux types de cartes, des cartes métriques aux représentations topologiques, avant d'aborder les approches modernes de cartographie sémantique et d'en analyser les limites. La section 2.3 décrit l'architecture du framework IntelliMap, en détaillant notamment la structure hiérarchique HSTM et les modalités d'intégration des données issues des capteurs. La section 2.4 expose le processus complet de construction d'une carte sémantique avec IntelliMap, incluant la génération de la carte métrique, la cartographie des objets, la segmentation des pièces et des bâtiments, ainsi que la catégorisation des lieux et la création des relations topologiques. Enfin, la section 2.5 conclut ce chapitre en synthétisant les apports de l'approche proposée.

2.2 État de l'art

2.2.1 les principaux types de cartes

Pour comprendre l'évolution des approches de cartographie, il est essentiel de définir les principaux types de cartes.

2.2.1.1 Cartes métriques

Une carte métrique est une représentation numérique de l'environnement qui encode des informations précises sur les distances, les angles et les positions relatives des objets dans l'espace. Ces cartes sont généralement basées sur un système de coordonnées (comme les coordonnées cartésiennes) et permettent une localisation précise des objets et des obstacles. Les premières cartes métriques ont été développées dans le contexte de la robotique mobile, où la nécessité de naviguer de manière autonome dans des environnements structurés a conduit à la création de représentations spatiales précises. Les travaux pionniers dans ce domaine [3, 4] ont introduit les cartes de grille d'occupation pour la navigation robotique. Avec l'avènement des systèmes de localisation et de cartographie simultanées, en anglais "Simultaneous Localization and Mapping (SLAM)", les cartes métriques sont devenues plus sophistiquées. Les robots peuvent désormais

construire des cartes en temps réel tout en se localisant dans ces cartes. Les travaux [5, 6] ont constitué une avancée majeure dans le domaine. Les progrès dans les capteurs et les algorithmes de traitement ont permis la création de cartes métriques de plus en plus précises et détaillées, utilisées dans des applications allant de la robotique domestique à la cartographie urbaine.

Dans le contexte de la navigation sémantique, les cartes métriques fournissent la base spatiale sur laquelle les informations sémantiques (comme les objets, les zones fonctionnelles, etc.) peuvent être superposées.

— Dimension des cartes métriques Les cartes métriques peuvent être classées selon leur dimension, en distinguant les cartes 2D des cartes 3D. Les cartes 2D représentent l'environnement dans un plan horizontal, généralement le plan XY, et sont couramment utilisées pour la navigation de robots mobiles dans des environnements intérieurs où la hauteur n'est pas un facteur critique. En revanche, les cartes 3D offrent une représentation volumétrique de l'environnement, incluant la hauteur (axe Z), ce qui les rend essentielles pour les applications nécessitant une compréhension de la structure verticale. Elles sont notamment utilisées dans des domaines comme la cartographie urbaine, l'inspection de structures verticales.

— Types de cartes métriques. Plusieurs types de cartes métriques sont couramment utilisées pour représenter l'environnement :



FIGURE 3 – Exemple de carte de grille d'occupation 2D générée à partir de l'étage 0 de la scène 00862 du dataset HM3DSem.

— Cartes de Grille d'Occupation : Les cartes de grille d'occupation divisent l'environnement en une grille régulière composée de cellules, où chaque cellule contient une probabilité d'occupation, généralement représentée par une valeur entre 0 (libre) et 1 (occupée). Cette approche probabiliste modélise efficacement les obstacles et les

espaces libres dans l'environnement. Les cartes de grille d'occupation sont largement utilisées en robotique pour la navigation et la planification de trajectoires, car elles offrent une représentation simple et intuitive de l'espace. Elles sont particulièrement adaptées aux environnements intérieurs et aux applications où la précision spatiale est cruciale, comme l'évitement d'obstacles ou la localisation de robots. Cependant, leur principale limitation réside dans leur coût en mémoire et en calcul pour les environnements de grande taille ou en 3D, car la résolution de la grille doit être suffisamment fine pour capturer les détails de l'environnement. L'absence de prise en compte de la hauteur des obstacles limite son utilisation à l'unique robot ayant effectué l'acquisition. Si le capteur est positionné à des hauteurs différentes, les cartes générées peuvent diverger, rendant la réutilisation du modèle par un autre robot complexe. Malgré cela, elles restent l'une des méthodes les plus populaires en robotique mobile. Un exemple concret de représentation d'un espace de bureau, est illustré à la figure 3.

— Nuage de points (Point Cloud) : Les nuages de points représentent l'environnement sous forme d'un ensemble de points en 3D générés à partir de capteurs tels que les LiDAR ou les caméras 3D, comme illustré sur la figure 4. Chaque point correspond à une mesure précise de la position d'une surface dans l'espace, capturant ainsi les détails fins et complexes de l'environnement. Les nuages de points peuvent présenter des difficultés pour la planification de trajectoire en raison de leur structure non segmentée, ne distinguant pas explicitement les zones navigables des obstacles. Cependant, leur densité élevée et leur croissance rapide avec la fréquence d'acquisition peuvent entraîner une consommation mémoire importante, en particulier dans des scénarios de navigation prolongés ou à haute résolution.

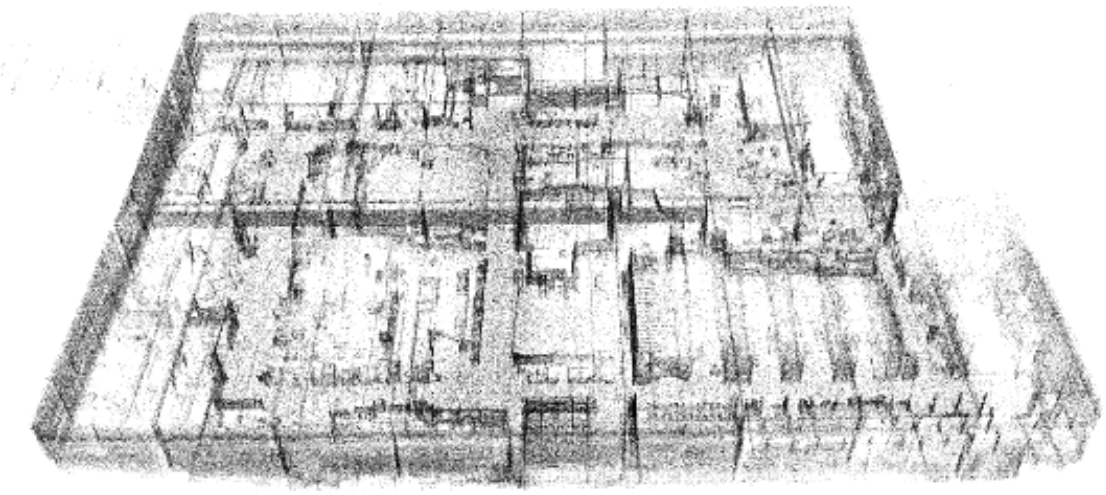


FIGURE 4 – Nuage de points 3D acquis dans un environnement industriel à l'aide du robot Scout équipé d'un LiDAR Hesai Pandar XT-32. Cette capture est réalisée dans une usine réelle.

— Cartes de Maillage (Mesh Maps) : Les cartes de maillage offrent une représentation 3D plus structurée que les nuages de points en utilisant un maillage polygonal pour

modéliser la surface des objets et des environnements. Contrairement aux nuages de points, qui sont une collection de points individuels, les maillages relient ces points pour former des triangles ou d'autres polygones, créant ainsi une surface continue et lisse. Cette approche permet une meilleure reconstruction de l'espace et une représentation plus compacte de l'environnement, comme illustré sur la figure 5. La génération de maillages peut être réalisée à partir de nuages de points, en appliquant des algorithmes de reconstruction de surface. Elle peut également s'appuyer sur des données issues de capteurs RGB-D ou LiDAR, qui fournissent des informations géométriques denses sur l'environnement. Ces techniques nécessitent cependant un traitement géométrique avancé pour assurer une reconstruction précise, continue et cohérente des surfaces. Sur le plan mémoire, les cartes de maillage peuvent s'avérer coûteuses, en particulier lorsque le niveau de détail est élevé, car elles stockent non seulement les positions des sommets, mais aussi la connectivité topologique entre ces sommets (arêtes, faces), ce qui augmente significativement le volume de données à conserver.

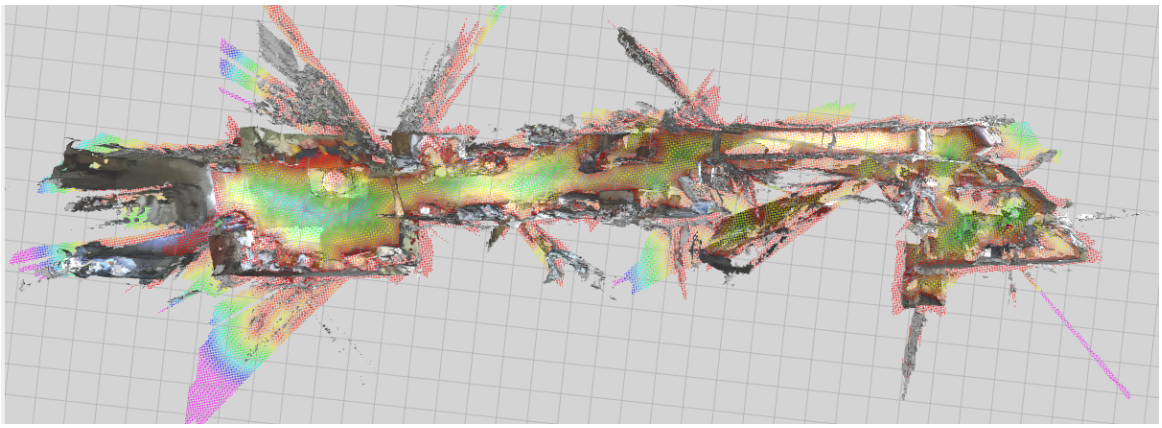


FIGURE 5 – Carte de maillage du premier étage des anciens bureaux de l'entreprise euroDAO.

Dans une approche de navigation sémantique, les cartes ne se limitent pas à la simple représentation spatiale. Elles servent de support à l'intégration d'informations sémantiques, permettant d'associer des objets à des emplacements spécifiques sur la carte, voire à des zones fonctionnelles de l'environnement, et ainsi d'améliorer la planification des trajets. Cette association peut s'appuyer sur des informations géométriques extraites de la carte métrique (comme les contours ou la position des murs) pour déduire des relations spatiales pertinentes. Par exemple, les contours géométriques d'une pièce peuvent être utilisés pour associer un four à une cuisine ou un lit à une chambre. Une carte de grille d'occupation peut être enrichie par des labels sémantiques pour identifier les zones accessibles et leur fonction. Un nuage de points peut être enrichi par des informations de couleur associées à des classes sémantiques, afin d'identifier et de catégoriser les objets présents dans l'environnement. Une carte de maillage peut être utilisée pour segmenter des espaces complexes et attribuer des relations fonctionnelles aux structures. Ainsi, les

cartes métriques constituent l'infrastructure sous-jacente sur laquelle s'appuie la navigation sémantique, en offrant une modélisation robuste et évolutive de l'environnement.

2.2.1.2 Cartes topologiques

Les cartes topologiques constituent une approche alternative à la modélisation métrique de l'environnement, privilégiant une représentation structurée sous forme de graphe. Dans ce modèle, l'espace est segmenté en lieux significatifs, tels que des pièces, des intersections ou des zones fonctionnelles, qui sont représentés sous forme de nœuds. Ces nœuds sont reliés par des arêtes, qui traduisent les connexions existantes entre les différents lieux et peuvent être enrichies par des informations sur les actions nécessaires pour les traverser.

Contrairement aux cartes métriques, qui nécessitent une localisation précise du robot dans un référentiel global, les cartes topologiques s'appuient sur des relations spatiales qualitatives, sans nécessiter de coordonnées exactes. Cette abstraction réduit la complexité du modèle tout en offrant une représentation compacte, adaptée aux environnements de grande dimension. De plus, en limitant le nombre d'informations analysées, les cartes topologiques optimisent la planification de trajectoires et facilitent la navigation à haut niveau, rendant les prises de décision plus rapides et robustes dans des environnements complexes.

Cependant, la création d'une carte topologique repose sur la capacité du robot à identifier des lieux significatifs, ce qui peut être difficile dans des environnements non structurés ou inconnus. Dans des environnements très dynamiques, où la connectivité entre les lieux change fréquemment, les cartes topologiques peuvent nécessiter des mises à jour fréquentes. Dans le contexte de la navigation sémantique, les cartes topologiques peuvent être enrichies avec des annotations sémantiques, associant chaque nœud ou arête à des informations sémantiques contextuelles.

En se concentrant sur les relations spatiales et en intégrant des annotations sémantiques, elles permettent aux robots de naviguer de manière plus intelligente et adaptative. Bien qu'elles soient utiles, les cartes topologiques présentent certaines limites, notamment en précision géométrique. Elles décrivent les connexions entre lieux, mais sans représenter fidèlement les distances ou les formes. Elles n'intègrent pas non plus les obstacles locaux. De plus, leur construction demande souvent une exploration presque complète de l'environnement. Malgré cela, leur simplicité et leur capacité à intégrer des informations contextuelles en font un outil essentiel pour la navigation sémantique. Une illustration d'une carte topologique est présentée sur la figure 6. Combinées avec des cartes métriques et sémantiques, elles forment une base solide pour des robots autonomes capables de comprendre et d'interagir avec leur environnement de manière naturelle. Par exemple, [7] a proposé des cartes hybrides qui exploitent à la fois la précision des cartes métriques et la simplicité des cartes topologiques. Les cartes topologiques ont gagné en popularité avec l'émergence des systèmes de navigation basés sur des graphes comme [8].

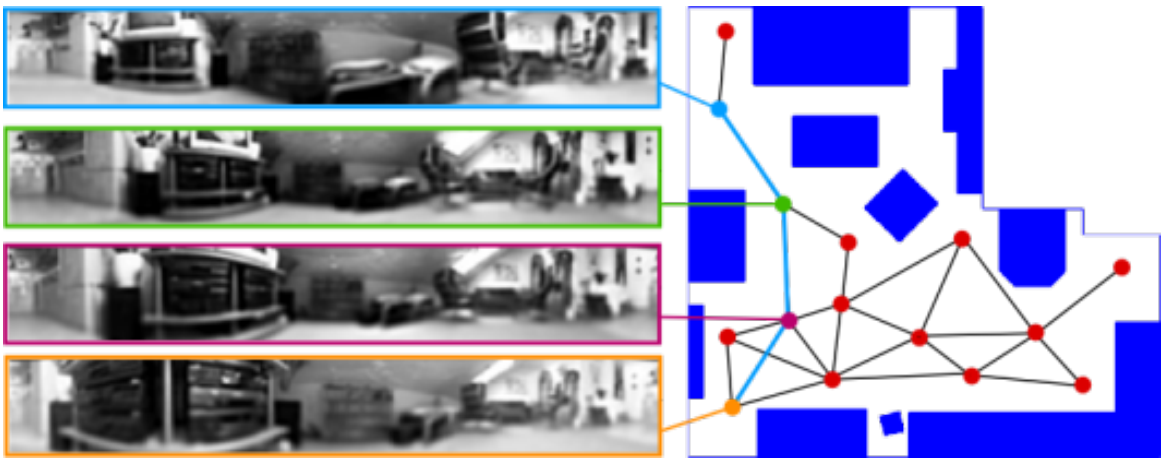


FIGURE 6 – Illustration du concept de carte topologique pour la navigation robotique. Chaque nœud représente un lieu caractérisé par des informations sensorielles (ex. image panoramique) collectées lors du premier passage du robot. Image obtenue depuis le site du laboratoire de l'Université de Bielefeld https://www.ti.uni-bielefeld.de/html/research/topmaps/index.html.

2.2.2 Approches modernes de cartographie sémantique

Les cartes sémantiques enrichissent les représentations métriques ou topologiques avec des informations contextuelles, telles que les types d'objets, les zones fonctionnelles (ex. cuisine, bureau) et les relations entre ces éléments. Elles permettent aux robots de comprendre et d'interagir avec leur environnement de manière plus intuitive, en allant au-delà de la simple détection d'obstacles pour intégrer une compréhension sémantique de l'espace. Cette compréhension sémantique est essentielle pour des tâches complexes comme la navigation contextuelle, l'interaction humain-robot (HRI) et la planification de tâches spécifiques (ex. "aller chercher un objet dans la cuisine"). Dans le contexte actuel de la recherche, deux catégories principales de cartes sémantiques se sont imposées : les cartes topométriques hiérarchiques et les graphes de scènes.

2.2.2.1 Cartes topométriques hiérarchiques

La première catégorie repose principalement sur des cartes topométriques hiérarchiques [9, 10, 1]. Les cartes topométriques hiérarchiques, inspirées de [9], constituent une structure multi-niveaux permettant de représenter l'environnement selon plusieurs abstractions complémentaires. Elles combinent des connexions topologiques (entre lieux comme pièces ou couloirs) avec des informations géométriques locales (trajectoires, directions, distances), tout en intégrant progressivement des éléments sémantiques à mesure que la connaissance de l'espace s'enrichit. Dans, [1] les auteurs proposent une méthode pour générer des cartes topométriques hiérarchiques à partir de nuages de points 3D. La figure 7, extraite de leur article, donne un exemple de ce type de carte.

Ces cartes sont construites à l'aide de méthodes d'inférence appliquées à des cartes métriques et sont dépourvues de détails sémantiques centrés sur les objets. Cette absence de détails sur

les informations sémantiques limite souvent leur capacité de reconnaissance à seulement trois catégories : les pièces, les couloirs et les portes. Bien que ces cartes soient opérationnellement efficaces, elles peuvent négliger l'importance des objets dans l'environnement. Par conséquent, la profondeur de l'extraction sémantique concernant les nuances spécifiques à chaque pièce peut être compromise, rendant ces cartes moins adaptées à des tâches nécessitant une compréhension fine de l'environnement, comme distinguer une cuisine d'une chambre à coucher. De plus, l'absence d'informations relatives aux objets peut considérablement appauvrir la richesse sémantique disponible lors des interactions homme-robot (HRI), limitant ainsi la compréhension contextuelle et l'adaptabilité du système.

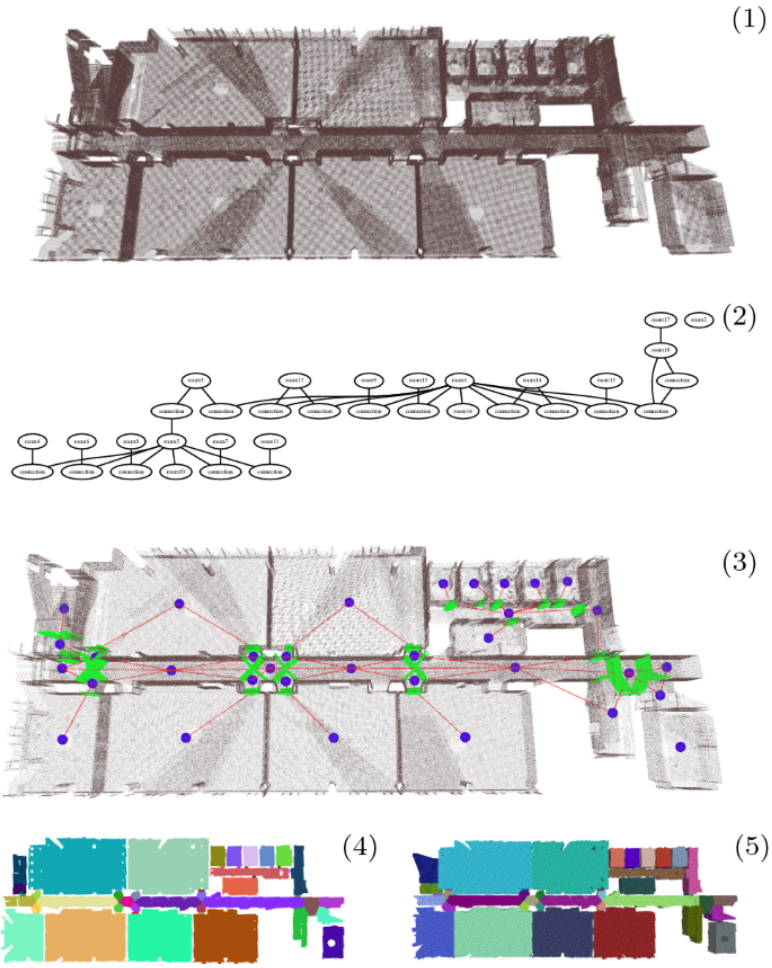


FIGURE 7 – À partir d'un nuage de points 3D en entrée (1), différentes représentations hiérarchiques sont générées : une carte topologique 0D (2), une carte 1D avec sommets annotés par leurs coordonnées (3), une carte 2D segmentée en régions colorées (4) et une carte 3D contenant des informations volumétriques (5). Image obtenue de [1].

2.2.2.2 Graphe de scène 3D (3D Scene Graph)

Pour améliorer ces cartes hiérarchiques sémantiques, une deuxième catégorie a émergé, utilisant des techniques de graphes de scène, connue sous le nom de graphes de scène 3D [11, 12, 13, 14, 15, 16]. Ces cartes offrent une représentation exhaustive de l'environnement en intégrant

aussi les objets détectés lors du processus de cartographie.

Parmi les exemples notables de ces travaux, on trouve [17, 18, 19]. Ces études présentent une représentation détaillée de l'environnement en incorporant plusieurs couches sémantiques : une couche de maillage 3D sémantique pour modéliser les surfaces et les structures ; une couche d'objets pour identifier et localiser les éléments individuels (ex. tables, chaises) ; une carte topologique complexe basée sur les zones d'espace libre ; une couche de pièces pour regrouper les objets et les zones fonctionnelles, et pour organiser les pièces et les étages.

Cette approche complète fournit un modèle métrique exhaustif de l'environnement, couplé à une compréhension sémantique multi-niveaux. Cependant, la complexité de ces cartes introduit également des défis importants. Bien que des efforts significatifs aient été réalisés dans des travaux comme [20] pour réduire la taille des cartes topologiques formées à l'aide de diagrammes de Voronoï généralisés (GVD) pour représenter les zones d'espace libre, ces cartes génèrent toujours un grand nombre de nœuds topologiques et nécessitent des ressources computationnelles importantes, en particulier dans des environnements à grande échelle. De plus, le processus de maillage demande une capacité de calcul considérable, ce qui rend ces cartes plus adaptées à des tâches de reconstruction 3D.

2.2.3 Limites des approches existantes

Chacun des types de cartes mentionnés présente ses avantages mais aussi ses propres limitations, soulignant la nécessité d'un type de carte qui équilibre efficacité, détails et compréhension sémantique.

Les cartes métriques, bien qu'elles offrent une représentation géométrique précise de l'environnement, peuvent être coûteuses en mémoire selon leur résolution, le volume de données stockées, et le type de représentation utilisé (par exemple, nuages de points, grilles d'occupation ou cartes maillées). Elles peuvent également être difficiles à enrichir sémantiquement, car elles ne structurent pas explicitement les entités fonctionnelles ou les objets. À l'inverse, les cartes topologiques sont compactes et facilitent la planification globale, mais souffrent d'un manque de précision géométrique et d'un encodage limité des relations sémantiques. Chacun de ces types de cartes présente donc des compromis entre expressivité, efficacité et pertinence contextuelle pour la navigation sémantique.

Les cartes topométriques hiérarchiques, bien que très efficaces pour la navigation à grande échelle grâce à leur structure simplifiée, peuvent rapidement présenter des limites en termes de détails sémantiques associés aux objets. En effet, leur granularité réduite, orientée vers la structuration topologique générale, ne permet pas toujours de capturer des informations sémantiques fines essentielles à certaines tâches spécifiques de navigation ou d'interaction.

À l'inverse, les graphes de scène 3D exploitent des représentations riches telles que des maillages, offrant une description très détaillée et visuellement pertinente de l'environnement. Cependant, ces représentations deviennent rapidement lourdes et coûteuses en calcul, notamment lorsqu'il s'agit de grands environnements composés de multiples étages ou couvrant de

vastes espaces. Par ailleurs, pour la navigation robotique, ces représentations complexes nécessitent généralement une conversion intermédiaire vers des structures simplifiées (par exemple, une costmap, c'est-à-dire une grille d'occupation de l'environnement où chaque cellule contient un coût de traversée, ou un graphe simplifié), ce qui entraîne une perte des bénéfices initiaux liés au détail visuel. Enfin, la navigation directe sur ces structures impose l'usage d'algorithmes spécialisés sur surfaces polygonales, qui demeurent plus coûteux et moins optimisés que les algorithmes classiques.

Dans ce contexte, IntelliMap propose une approche innovante pour répondre à ces limitations. En combinant la précision locale des cartes métriques, la simplicité structurelle des cartes topologiques et l'efficacité de la representation hiérarchique, ainsi que la richesse des annotations sémantiques, ce framework hiérarchique équilibre ces différents aspects. IntelliMap vise à fournir une représentation à la fois efficace et détaillée de l'environnement, adaptée aux applications en temps réel et aux environnements dynamiques.

2.2.4 Synthèse comparative : couverture de l'information sémantique

Une analyse comparative avec d'autres études de cartographie sémantique permet d'évaluer la richesse des informations couvertes par IntelliMap, telles que nous les présenterons dans ce chapitre. Dans ce contexte, le tableau 2.1 présente une comparaison d'IntelliMap par rapport à certaines méthodes existantes. Cette analyse examine plusieurs caractéristiques, notamment la représentation de la carte métrique, la couverture de la carte sémantique. Le tableau met en évidence qu'IntelliMap couvre une gamme plus large de caractéristiques par rapport aux autres méthodes étudiées. En particulier, IntelliMap utilise une costmap (pour rappel, une grille d'occupation où chaque cellule contient un coût de traversée) puis intégre de façon hiérarchique à la fois des informations relatives aux objets et aux pièces. Cette couverture étendue de l'information permet une compréhension plus fine de l'environnement, ce qui est essentiel pour des tâches complexes comme la navigation autonome dans des espaces dynamiques. Par exemple, la méthode de [21] utilise une Costmap, mais ne prend pas en compte les objets ni les relations sémantiques entre les différentes parties de l'environnement.

L'intégration de données sémantiques dans IntelliMap, avec sa structure hiérarchique adaptable, permet d'offrir une couverture plus riche et plus détaillée, notamment en représentant à la fois des objets et des pièces de manière précise, tout en tenant compte de leurs relations fonctionnelles et contextuelles.

Ainsi, IntelliMap se distingue par sa capacité à intégrer efficacement des informations provenant de plusieurs sources et à offrir une couverture sémantique plus complète.

Méthode	**Carte Métrique**	**Carte Sémantique**	
		Objet	**Pièce**
IntelliMap	Costmap	✓	✓
Zimmerman et al., 2023 [21]	Costmap	✓	✗
Hughes et al., 2022 [14]	Maillage (Mesh)	✓	✓
He et al., 2021 [1]	Nuage de points	✗	✓
Hiller et al., 2019 [10]	Costmap	✗	✓
Gemignani et al.,2016 [22]	Costmap	✓	✓
Paul et al.,2023 [23]	Contours de pièces + SDF	✓	✓
Devendra et al.,2020 [24]	Costmap	✓	✗

TABLE 2.1 – Analyse comparative des configurations de couches sur IntelliMap et d'autres méthodes existantes.

2.3 Framework IntelliMap : architecture

IntelliMap est un framework modulaire conçu pour générer des cartes sémantiques hiérarchiques adaptées aux différents environnements qu'ils soient complexes ou dynamiques. Il repose sur une architecture flexible qui fusionne différentes sources de données afin de structurer l'information spatiale de manière efficace et adaptable. Cette approche hiérarchique permet une représentation plus riche de l'environnement. Structuré autour de plusieurs modules interdépendants, IntelliMap combine des informations métriques, sémantiques et topologiques pour produire une représentation complète et adaptative de l'environnement.

Le framework IntelliMap, illustré dans la figure 8, s'appuie sur plusieurs modules complémentaires organisés en trois étapes : entrée, traitement, et sortie. À partir des données des capteurs, un module de SLAM métrique reconstruit la carte géométrique de l'environnement tout en estimant la position du robot. En parallèle, un module de cartographie des objets extrait les entités sémantiques détectées dans la scène.

Ces informations alimentent ensuite les processus d'organisation spatiale et de cartographie relationnelle, comprenant la segmentation des bâtiments, la segmentation des pièces et la catégorisation des lieux. Ce traitement structure les données en unités fonctionnelles et relations topologiques pertinentes.

En sortie, le système produit une carte métrique sémantique et une carte topologique sémantique, qui sont combinées pour générer une Hierarchical Semantic Topometric Map (HSTM). La HSTM combine des informations métriques, sémantiques et topologiques pour une navigation

robotique adaptative. Les sections suivantes détaillent la structure de la HSTM, les différents modules et leur rôle dans le processus de génération de la carte sémantique.

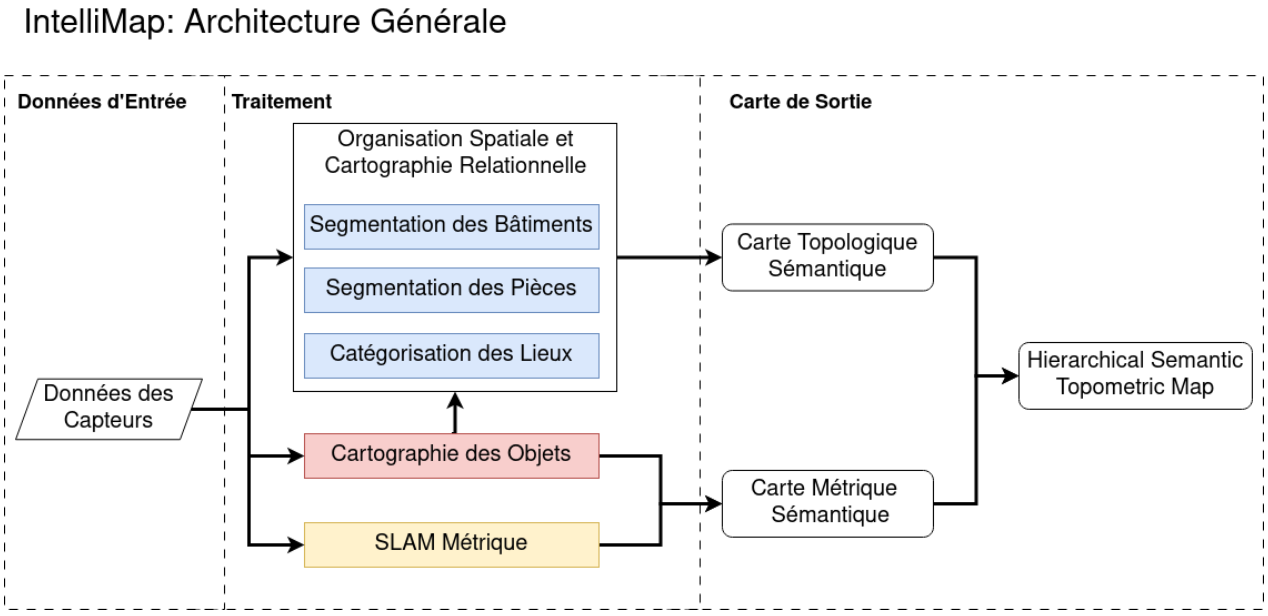


FIGURE 8 – Schéma de l'architecture du framework IntelliMap. Ce schéma illustre les principaux modules du framework IntelliMap, conçu pour générer une carte HSTM à partir de données capteurs brutes. Le processus comprend la segmentation des pièces et des bâtiments, la catégorisation des lieux, la cartographie des objets et la SLAM métrique. Les résultats intermédiaires, carte topologique sémantique et carte métrique sémantique, sont fusionnés pour produire une représentation hiérarchique intégrant les aspects topologiques, sémantiques et métriques de l'environnement.

2.3.1 Structure de la HSTM

Le framework IntelliMap produit en sortie une carte hiérarchique HSTM, qui constitue le résultat final du processus de cartographie. La HSTM sert de référence pour la navigation autonome du robot, en intégrant des détails précis sur la disposition de l'espace, les objets présents et les relations fonctionnelles entre les différentes zones.

Dans notre implémentation, les couches objets et pièces sont représentées conjointement dans un même fichier JSON. La couche métrique, quant à elle, est stockée séparément.

Ces fichiers structurés sont utilisés à la fois pour enregistrer les données extraites lors de la phase de cartographie et pour un chargement dynamiquement lors de la navigation afin de fournir au robot les informations nécessaires à la planification, à la prise de décision et à la visualisation de la carte.

Nous nous sommes inspirés du travail présenté dans la référence [19] pour structurer les couches sémantiques de la HSTM sous forme de graphe enregistré dans un fichier JSON.

2.3.1.1 Couche métrique

La couche métrique constitue la base de la hiérarchie et fournit une description détaillée de la géométrie de l'environnement. Elle est générée via un module de SLAM métrique (Simultaneous Localization and Mapping) qui capture la disposition spatiale en termes de distances, d'obstacles et de contours. Cette couche est enregistrée indépendamment dans des fichiers dont le format dépend de sa nature, tels que .pcd pour les nuages de points 3D, .png ou d'autres formats d'image pour les grilles d'occupation (i.e. costmap).

Cette couche est indispensable pour la navigation locale, l'évitement d'obstacles et la planification de trajets précis. La couche métrique fournit une référence absolue. Celle-ci sert au robot pour se localiser précisément et effectuer des corrections de trajectoire en temps réel lors de son déplacement. Elle sert également de fondation pour l'intégration des couches sémantiques, garantissant une cohérence entre les informations géométriques et contextuelles.

2.3.1.2 Couche des objets

IntelliMap enrichit la HSTM en introduisant une couche dédiée aux objets, représentant explicitement les entités détectées dans l'environnement. Cette couche est constituée d'un ensemble de nœuds, chacun correspondant à un objet identifié, classé et localisé à partir des données issues des capteurs. Ces nœuds sont enregistrés dans le fichier JSON, où chaque objet est décrit par un ensemble d'attributs. La structure inclut les éléments suivants, tout en restant extensible afin d'intégrer, si nécessaire, des informations supplémentaires en fonction des exigences spécifiques des scénarios de navigation.

- ID : Un identifiant unique pour l'objet.
- Position : Coordonnées 3D du centre géométrique de l'objet, exprimées dans le repère monde. Il sert à positionner l'objet sur une carte ou à afficher un label.
- Bounding Box : Une boîte englobante définissant les limites de l'objet.
- Semantic Label : Identifiant sémantique (par exemple une classe d'objet dans un dataset)
- Color : Une couleur associée à l'objet pour en faciliter la visualisation, définie en fonction des catégories sémantiques issues du modèle de détection utilisé.
- Weight [1] : Booléen reflétant la présence effective de l'objet à sa position attendue. Il permet de refléter dynamiquement l'état de présence de l'objet et d'adapter la prise de décision en conséquence.

L'exemple ci-dessous montre comment un objet réel, ici une porte, est représenté dans le format JSON avec ses attributs spécifiques :

```
{
    "id": 389,
    "layer": 2,
    "attributes": {
```

1. Attribut Booléen indiquant si l'objet est toujours détecté à sa position enregistrée pendant la mission en cours.

```
        "name": "door",
        "position": [
                    1.8368055820465088,
                    1.0456063747406006,
                    -2.5344595909118652
                ],
        "bounding_box": {
            "type": "RAABB",
            "max": [
                        2.2968130111694336,
                        2.0914039611816406,
                        -2.439578056335449
                    ],
            "min": [
                        1.376798152923584,
                        -0.00019121170043945312,
                        -2.6293411254882812
                    ],
        },
    "semantic_label": 389,
    "color": [228.0, 105.0, 12.0]
    "weight": false
    }
}
```


Dans cet exemple, la propriété “layer” : 2 indique que l’objet appartient à la couche objets de la carte sémantique hiérarchique. Comme expliqué précédemment, cette couche constitue le deuxième niveau du modèle, venant après la couche métrique de base (layer = 1), qui représente la structure géométrique de l’environnement. Ces objets peuvent être intégrés directement à la carte métrique sous forme d’annotations ou être stockés dans une couche distincte pour une gestion plus flexible.

L’identification des objets repose sur des modèles de vision artificielle et d’apprentissage automatique qui permettent au robot de reconnaître les éléments de son environnement en temps réel. Le processus est détaillé dans la section 2.4.3.

2.3.1.3 Couche des pièces et des bâtiments

Dans un environnement intérieur, il est essentiel que le robot puisse comprendre la segmentation fonctionnelle de l’espace afin d’adopter des stratégies de navigation adaptées. La couche des pièces segmente tout d’abord l’environnement en zones distinctes, telles que des pièces ou des aires de travail, en s’appuyant sur des indices structurels (murs, portes, passages) et des indices sémantiques liés aux objets présents.

Chaque pièce est représentée par un nœud dans le fichier JSON, avec des attributs tels que :

— ID : Un identifiant unique pour la pièce, utilisé pour établir ses relations avec les autres entités de la carte.

— Bounding Box [2] : Une boîte englobante approximative servant principalement à la visualisation de l'emplacement général de la pièce. Elle est centrée sur le centroïde de la pièce et ne définit pas ses limites exactes.

— Polygon 2D [3] : Un polygone en deux dimensions qui décrit la forme et l'étendue réelle de la pièce, utilisé pour la segmentation spatiale et l'association avec les objets présents.

— Semantic Label [4] : Une étiquette sémantique indiquant la catégorie fonctionnelle de la pièce (ex. "salon", "cuisine").

— Weight [5] : Un poids dynamique utilisé pour la priorisation lors de la navigation ou de l'exploration.

— Floor Level [6] : Un identifiant indiquant le niveau de l'étage auquel appartient la pièce. Il permet une gestion efficace des transitions intra-étage et inter-étages.

L'exemple de nœud ci-dessous correspond à une pièce dans le fichier JSON correspondant à une HSTM d'un environnement. Les relations spatiales entre les pièces et les objets seront modélisées sous forme d'arêtes, qui seront définies séparément.


```
{ "id": 300,
  "layer": 3,
  "attributes": {
      "name": "living_room",
      "position": [
          7.6416359076555604,
          -0.16965234857424596,
          -4.246532642203577
          ],
      "bounding_box": {
          "type": "RAABB",
          "min": [
              5.0006366001767795,
              -0.13212442081313802,
              -1.3832588835160449
              ],
          "max": [
              9.57821040539695,
              -0.13212442081313638,
```


2. Terme standard en vision par ordinateur, désignant une boîte englobante autour d'un objet ou d'une zone.
3. Forme géométrique décrivant les contours de la pièce dans un plan 2D.
4. Étiquette sémantique indiquant la fonction de la pièce, comme "salon" ou "cuisine".
5. Booléen indiquant si la pièce a été visitée lors de la mission en cours. `true` = visitée, `false` = non visitée.
6. Identifiant d'étage permettant de distinguer les pièces appartenant à différents niveaux dans une scène multi-étages.

```
                3.761175807219838
                ]
        },
        "polygon_2d": [
            [0,0],
            ...,
            [0,0],
            [0,0]
        ],
    "semantic_label": 100,
    "color": [0.0, 0.0, 0.0]
    "node_weight": false,
    "floor_level": 0,
    }
}
```

Les relations entre les pièces et les objets sont modélisées par des arêtes dans le fichier JSON. Il existe deux types d'arêtes :

- Arêtes entre pièces (room-to-room edges) : Ces arêtes relient deux nœuds de pièces et indiquent que l'on peut traverser de l'une à l'autre ("traversabilité"). Elles modélisent les connexions spatiales directes et peuvent inclure des informations supplémentaires telles que la relation topologique verticale (étages différents) ou horizontale (même étage).
  - Source : Identifiant unique du nœud source (ID de la pièce d'origine).
  - Target : Identifiant unique du nœud cible (ID de la pièce connectée).
  - Edge_link : Indique le type de connexion entre les pièces. Par exemple : "vertical" : les pièces sont situées à des niveaux différents (ex. escalier, ascenseur) ; "none" : les pièces sont situées sur le même niveau.
  - Waypoints (pour les connexions verticales uniquement) : Liste ordonnée de positions 3D intermédiaires que le robot doit suivre pour effectuer la transition verticale entre les deux pièces connectées. Ces waypoints permettent une navigation entre les niveaux en tenant compte de la géométrie réelle du bâtiment.
  - Weight : Représente le coût dynamique de traversabilité entre deux pièces connectées. Ce poids est mis à jour dynamiquement en fonction de l'état du chemin correspondant. Il permet de guider la planification sémantique en favorisant les connexions sûres et en pénalisant les chemins jugés risqués ou bloqués.

L'exemple d'arête (edge) ci-dessous modélise la relation spatiale entre les pièces 3008 (un salon) et 3007 (un couloir) situés à deux niveaux différents. L'attribut "weight" représente un coût associé au passage entre les deux pièces, ici égal à 4. Ce poids est utilisé lors de la navigation pour pondérer les chemins.

```
        "edges": [
```

```
            {
                "source": 3008,
                "target": 3017,
                "info": {
                    "type": "EdgeAttributes",
                    "weight": 4,
                    "edge_link": "none"
                }
            }]
```

— Arêtes entre pièces et objets (room-to-object edges) : Ces arêtes relient une pièce à un objet, indiquant que l'objet est situé dans cette pièce. L'exemple ci-dessous modélise une arête (edge) représentant la relation entre une pièce et un objet. Ces arêtes indiquent qu'un objet spécifique est situé à l'intérieur d'une pièce donnée.

```
        "edges": [
            {
                "source": 3000,
                "target": 4,
                "info": {
                    "type": "EdgeAttributes",
                }
            }
        ]
```

Cette couche permet une navigation plus contextuelle, où le robot peut raisonner en termes d'espaces fonctionnels plutôt qu'en simples coordonnées spatiales. Par exemple, un robot cherchant un four à micro-ondes pourra privilégier la cuisine plutôt que d'explorer aléatoirement l'ensemble de l'environnement. Dans les environnements comportant plusieurs bâtiments, une couche supplémentaire est introduite pour différencier explicitement ces structures. Cette organisation est assurée par la couche bâtiment, présentée ci-après.

La couche des pièces peut comprendre également une description de bâtiments ou sous-bâtiments. En effet, dans certains environnements complexes, comme les grands bâtiments industriels, les hôpitaux, ou les campus universitaires, une seule entité structurelle peut être divisée en plusieurs sous-bâtiments distincts (ex. Bâtiment A, Bâtiment C). Cette séparation n'implique pas nécessairement une discontinuité physique stricte, mais peut résulter d'une organisation spatiale et fonctionnelle (ex. zones de soins vs. zones administratives), afin de refléter la logique d'usage des espaces.

Souvent, ces sous-bâtiments sont interconnectés par des couloirs, des passerelles ou des zones de transition qui permettent le passage fluide d'un espace à un autre. Par conséquent, la représentation cartographique sémantique doit tenir compte de ces connexions tout en maintenant une distinction entre les différentes sections du bâtiment.

Par exemple, un site industriel peut être composé de plusieurs bâtiments interconnectés,

comme un bâtiment administratif et un atelier de production. Le bâtiment administratif regroupe des bureaux, des salles de réunion et des salles de contrôles tandis que l'atelier de production comprend les chaînes d'assemblage, les zones de stockage, les espaces de maintenance et les vestiaires. Ces bâtiments sont souvent reliés par des couloirs ou des voies logistiques, facilitant le transport de matériel et le déplacement du personnel.

Dans IntelliMap, ces relations sont modélisées par des arêtes entre les nœuds des sous-bâtiments dans la couche des pièces, représentant les connexions accessibles. Cela permet au robot d'interpréter ces connexions et d'adapter sa stratégie de navigation en conséquence, par exemple en prenant un couloir reliant le bâtiment A au bâtiment B, sans nécessairement considérer qu'il s'agit d'une transition entre deux entités complètement séparées. Elle est particulièrement utile pour la planification de trajets à grande échelle, où le robot doit naviguer entre plusieurs entités structurelles reliées.

Chaque bâtiment est représenté par un nœud dans le fichier JSON, avec des attributs tels que :

— ID : Un identifiant unique associé à chaque bâtiment, utilisé pour le différencier dans la structure de la carte.

— Semantic Label : Une étiquette sémantique décrivant la fonction du bâtiment (ex. "Bâtiment A", "Bâtiment B", "Atelier de production").

— Bounding Box : Une boîte englobante utilisée uniquement à des fins de visualisation, centrée autour du point moyen du bâtiment. Elle ne reflète pas l'étendue géométrique réelle du bâtiment, mais sert à représenter visuellement le nœud sur la carte.

L'exemple ci-dessous présente la modélisation d'un Bâtiment.


```
    {
  "id": 400,
  "layer": 4,
  "attributes": {
      "name": "Batiment A",
      "bounding_box": {
          "type": "RAABB",
          "min": [
              49.75,
              24.75,
              3.75
              ],
          "max": [
              50.25,
              25.25,
              4.25]
      },
      "semantic_label": 400
  }
```

```
}
```

Les relations entre les bâtiments et les pièces sont modélisées par des arêtes dans le fichier JSON. Il existe deux types d'arêtes pour la couche bâtiment :

— Arêtes entre bâtiments (Building-to-Building Edges) : Ces arêtes relient deux nœuds de bâtiments, indiquant une connexion physique ou logique entre eux (ex. bâtiments annexes reliés par une passerelle).

— Arêtes entre bâtiments et pièces (Building-to-Room Edges) : Ces arêtes relient un bâtiment à une pièce, indiquant que la pièce appartient à ce bâtiment.

Nous montrons ci-dessous un exemple de structure JSON pour une arête entre un bâtiment et une pièce ou bâtiment :

```
    "edges": [
        {
            "source": int,
            "target": int,
            "info": {
                "type": "string",
                "weight": int
            }
        }
    ]
```

La distinction entre les différents types de nœuds (par exemple, bâtiment ou pièce) est réalisée par inférence en analysant la nature des nœuds reliés par l'arête. Le type de chaque nœud étant encodé dans sa propre définition, cette information permet d'interpréter sémantiquement l'arête.

Cette structuration hiérarchique confère à IntelliMap une modularité optimale. La carte peut ainsi être adaptée aux besoins spécifiques de chaque scénario de navigation. Le format JSON permet une gestion flexible des données, facilitant les mises à jour dynamiques et l'adaptation à des environnements complexes. Cette architecture garantit un équilibre entre richesse sémantique et efficacité computationnelle, tout en facilitant les mises à jour dynamiques en réponse aux changements de l'environnement.

2.4 Processus de construction d'une carte avec IntelliMap

Le framework IntelliMap, illustré par la figure 8, repose sur une méthodologie structurée pour la génération de cartes sémantiques hiérarchiques (HSTM). La construction de cette carte suit une démarche modulaire et hybride, où la majorité des étapes, telles que la cartographie métrique, la cartographie des objets et les suivantes sont réalisées automatiquement à l'aide

d'outils de perception. Ce processus s'effectue hors ligne, avec la possibilité d'appliquer ponctuellement des corrections manuelles, notamment en cas de sur-segmentation.

L'une des particularités du framework IntelliMap réside dans sa flexibilité : les traitements associés à chaque couche (métrique, des objets, des pièces) peuvent être générés séparément et adaptés selon les besoins spécifiques de navigation. Ainsi, différentes méthodologies peuvent être employées pour construire chaque niveau de la carte, en fonction des contraintes de l'application visée. Dans les sections suivantes, nous détaillons les méthodes concrètement utilisées pour la génération de chaque composant du framework.

2.4.1 Données des capteurs : acquisition et intégration

Pour assurer une cartographie robuste et précise, IntelliMap exploite plusieurs sources de données issues de capteurs variés. Ces capteurs permettent de capturer à la fois des informations géométriques, sémantiques et contextuelles.

— Caméras : Les caméras fournissent des données visuelles essentielles pour la reconnaissance d'objets, l'identification de textures et l'analyse des scènes. Elles permettent notamment la reconnaissance et l'identification des objets présents dans l'environnement et l'enrichissement de la carte métrique avec des annotations sémantiques. Dans le cadre d'IntelliMap, nous exploitons des caméras RGB-D, qui fournissent à la fois une image couleur (RGB) et une carte de profondeur (D). Ces caméras permettent de capturer des informations 3D, essentielles pour la détection d'objets et la construction de la carte sémantique.

— LiDAR : Fournit des mesures de distance précises pour générer une carte métrique détaillée. Le terme LiDAR est un acronyme anglais pour "LIght Detection And Ranging". Dans le cadre d'IntelliMap, le LiDAR est utilisé pour construire une carte métrique de base, offrant une précision géométrique supérieure à celle des données de profondeur fournies par une caméra RGB-D. Cette carte sert de référence pour la localisation et la planification des trajectoires. Il est particulièrement important en navigation locale, où il permet au robot d'ajuster ses déplacements en fonction des obstacles proches et des variations dynamiques de l'environnement. En complément des données des caméras RGB-D, le LiDAR améliore la perception fine de l'espace, renforçant ainsi la précision et la robustesse de la navigation sémantique.

— IMU (Unités de Mesure Inertielle) : Il mesure l'accélération, la vitesse angulaire et son orientation à l'aide d'une combinaison d'accéléromètres, de gyroscopes et de magnétomètres. En complément, il permet d'ajouter une précision accrue lors du processus de cartographie, en améliorant la cohérence entre les données des différents capteurs.

2.4.2 SLAM Métrique : génération de la Carte Métrique

Dans le module de SLAM métrique d'IntelliMap, nous nous appuyons sur les nombreuses avancées du domaine de la cartographie métrique. Plusieurs algorithmes SLAM existent pour

générer différentes représentations métriques de l'environnement, qu'il s'agisse de cartes de grille d'occupation ou de nuages de points. Cette flexibilité permet d'adapter le choix de l'algorithme en fonction des objectifs spécifiques de navigation du robot.

Les cartes de grille d'occupation constituent l'une des représentations les plus utilisées en navigation robotique. Elles sont particulièrement adaptées aux environnements intérieurs. Parmi les algorithmes les plus couramment utilisés pour générer ces cartes, Gmapping [25, 26], basé sur un filtre de particules, génére des cartes 2D à partir de données LiDAR et est largement intégré dans des frameworks comme ROS (Robot Operating System) [27]. ROS constitue une infrastructure logicielle modulaire qui facilite le développement et l'intégration de composants robotiques hétérogènes en fournissant des outils, des bibliothèques et des interfaces standardisées. Son adoption généralisée dans la communauté robotique s'explique notamment par la facilité qu'il offre pour l'interopérabilité entre capteurs, algorithmes de SLAM, systèmes de visualisation (comme RViz) et modules de contrôle, ce qui en fait une plateforme de référence pour la recherche et le prototypage. Une autre approche, Hector SLAM [28], est adaptée aux environnements structurés et ne nécessite pas de données odométriques. Ces méthodes sont principalement appliquées à la navigation de robots mobiles en intérieur et à la planification de trajectoires. Les algorithmes de SLAM disponibles dans ROS peuvent être utilisés pour générer des cartes de grille d'occupation, qui servent ensuite de base à la construction de costmaps. Une costmap (carte de coûts) est une carte dérivée qui assigne un coût de traversabilité à chaque cellule, en fonction de sa proximité avec des obstacles ou d'autres critères de sécurité. Elle est essentielle pour la planification locale de trajectoires, car elle permet au robot d'identifier les zones à éviter ou à privilégier.

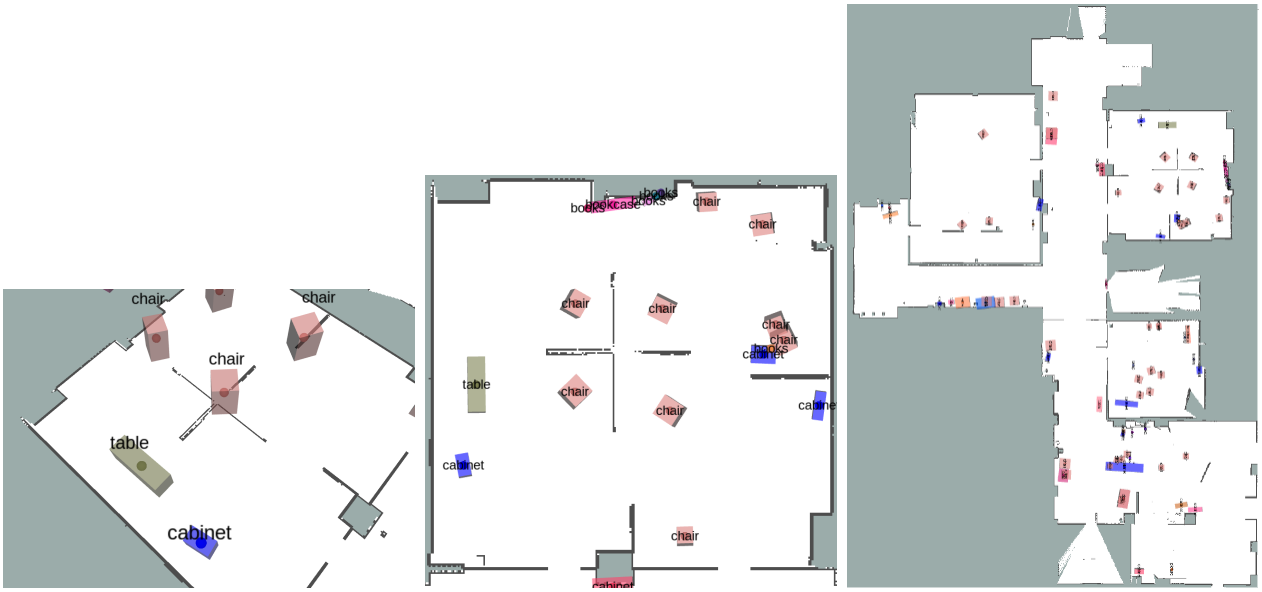
En parallèle, les nuages de points sont une représentation tridimensionnelle de l'environnement, constituée d'un ensemble de points discrets capturés par des capteurs comme le LiDAR ou les caméras 3D. L'exploitation des nuages de points repose sur différentes technologies avancées de cartographie 3D. Parmi elles, LOAM (LiDAR Odometry and Mapping) [29] est une approche efficace qui réalise simultanément l'odométrie et la cartographie en temps réel en utilisant les données issues d'un LiDAR. Une autre solution fréquemment utilisée est RTAB-Map (Real-Time Appearance-Based Mapping) [30], ce framework est compatible avec des caméras RGB-D et des capteurs LiDAR. Il prend en charge la cartographie 3D détaillée avec des dispositifs tels qu'une Kinect ou une caméra stéréo, ainsi qu'une cartographie 2D ou 3D lorsqu'il est utilisé sur un robot équipé d'un télémètre laser.

En complément, des méthodes plus avancées comme [31, 32] permettent de générer des nuages de points plus précis en combinant des données LiDAR, IMU et GPS (en français : Système mondial de positionnement). Ces approches sont aussi efficaces pour la navigation en grands espaces.

Grâce à cette diversité d'algorithmes SLAM, IntelliMap peut s'adapter aux besoins de navigation, que ce soit pour une cartographie 2D basée sur une grille d'occupation ou une cartographie 3D en nuage de points en fonction des capteurs disponibles et des contraintes de l'environnement robotique.

2.4.3 Cartographie des objets

La couche des objets dans l'architecture d'IntelliMap est conceptualisée comme un ensemble structuré de boîtes englobantes 3D $bbox3D$, noté Ω, où chaque élément $\omega \in \Omega$ est défini par une boîte englobante tridimensionnelle associée à une étiquette sémantique comme illustré sur la figure 9. Cette couche est développée à travers deux processus fondamentaux : la détection 3D et la cartographie des objets, essentiels pour améliorer la compréhension sémantique et la perception spatiale de l'environnement ciblé.



(a) Zoom sur un sous-ensemble d'objets dans une pièce (b) Ensemble des objets d'une pièce complète (c) Carte complète de l'environnement avec tous les objets

FIGURE 9 – Visualisations sémantiques des objets détectés dans l'environnement d'un bureau extrait du dataset UHumans2.

2.4.3.1 Détection d'objets

Chaque objet détecté ω est représenté sous la forme :

$$\omega = (l, C, D) \tag{2.1}$$

où :

- l est l'étiquette de l'objet, fournissant une classification catégorielle (par exemple, « table », « chaise », « armoire »).
- $C = (x, y, z)$ représente la position du centre de la $bbox3D$ dans l'espace 3D.
- $D = (w, h, d)$ dénote les dimensions de la $bbox3D$.

La détection des objets dans l'architecture IntelliMap repose sur Omni3D, un modèle avancé conçu pour la reconnaissance d'objets en 3D à partir d'une seule image [33]. Omni3D est entraîné sur un vaste jeu de données regroupant plusieurs ensembles préexistants tels que SUN RGB-D [34], KITTI [35] et nuScenes [36], totalisant plus de 234 000 images annotées avec

plus de 3 millions d'instances réparties sur 98 catégories d'objets. Omni3D exploite un réseau de neurones convolutifs appelé Cube R-CNN, capable de détecter des boîtes englobantes 3D ($bbox3D$) directement à partir d'une image 2D. Ce modèle généraliste fonctionne sur une large diversité de scènes, qu'il s'agisse d'environnements intérieurs (bureaux, maisons) ou extérieurs (routes, parkings), en générant des informations sur la localisation, l'orientation et la taille des objets détectés. Un des défis majeurs du passage de la 2D à la 3D est l'ambiguïté liée à la profondeur et aux variations de focales des caméras utilisées dans différents jeux de données. Pour y remédier, Omni3D introduit le concept de profondeur virtuelle, qui normalise les informations de profondeur afin d'uniformiser l'apprentissage et d'améliorer la robustesse du modèle.

En intégrant Omni3D dans IntelliMap, nous obtenons une détection initiale des objets ainsi qu'une estimation de leur position comme illustré dans la figure 10, fournissant ainsi une base exploitable pour construire la couche des objets à l'aide de traitements complémentaires.



FIGURE 10 – Exemple de détection d'objet avec Omni3D dans le couloir de l'environnement Office du dataset UHuman2. L'objet détecté est une chaise (chair), encadrée par une boîte englobante rose.

2.4.3.2 Vérification de l'unicité des objets

Pour garantir l'unicité des objets dans la couche, nous implémentons un processus de vérification pour chaque nouvelle $bbox3D$ détectée (notée $bbox3D_1$ dans l'algorithme présenté ci-dessous). Ce processus consiste à la comparer avec les $bbox3D$ existantes dans la couche (représentées par $bbox3D_2$ en utilisant une fonction de similarité, comme décrit dans l'algorithme 1.

L'algorithme AreSimilar vérifie si deux objets peuvent être considérés comme identiques en tenant compte de trois critères :

— **La proximité spatiale**, évaluée à partir de la distance euclidienne entre les centres des *bbox*3*D* : ($\text{dist}_{\text{ctr}} = \text{EUCLIDEAN}(\text{ctr}_1, \text{ctr}_2)$). Si cette distance est inférieure à un seuil $\text{dist}_{\text{thresh}}$, les objets sont considérés comme proches.

— **La similarité dimensionnelle**, mesurée via la différence absolue entre les tailles des deux boîtes ($\text{diff}_{\text{dim}} = \text{ABSDIFF}(\text{dim}_1, \text{dim}_2)$). Cette différence doit rester inférieure à un seuil $\text{size}_{\text{thresh}}$ pour être considérée négligeable.

— **La cohérence sémantique**, c'est-à-dire la comparaison des étiquettes (lbl_1, lbl_2). Deux objets sont plus susceptibles d'être considérés comme identiques s'ils portent la même étiquette.

L'algorithme retourne `True` (objets similaires) dans les cas suivants :

— Les objets ont une étiquette identique et sont suffisamment proches spatialement.

— Même si les étiquettes diffèrent, ils sont considérés comme similaires si leur distance est inférieure au seuil et leurs dimensions sont suffisamment proches.

Sinon, `False` est retourné, et la nouvelle boîte est ajoutée comme un nouvel objet dans la couche.

Cette approche permet de limiter la duplication d'objets en compensant les imprécisions des modèles de détection, telles que les erreurs d'étiquetage ou les fluctuations dans la position ou les dimensions détectées. Elle garantit ainsi une représentation plus cohérente et compacte de l'environnement.

Algorithm 1 Vérification de l'Unicité des Objets

Require: $\text{bbox3D}_1, \text{bbox3D}_2, \text{lbl}_1, \text{lbl}_2, \text{dist}_{\text{thresh}}, \text{size}_{\text{thresh}}$
Ensure: **True** si les objets sont similaires, **False** sinon

procedure AreSimilar
  $\text{dist}_{\text{ctr}} \leftarrow \text{EUCLIDEAN}(\text{ctr}_1, \text{ctr}_2)$
  $\text{diff}_{\text{dim}} \leftarrow \text{ABSDIFF}(\text{dim}_1, \text{dim}_2)$
  if ($\text{lbl}_1 \neq \text{lbl}_2$ **or** $\text{lbl}_1 = \text{lbl}_2$) **and** $\text{dist}_{\text{ctr}} < \text{dist}_{\text{thresh}}$ **and** $\text{ALL}(\text{diff}_{\text{dim}} < \text{size}_{\text{thresh}})$ **then**
    return True
  else if $\text{lbl}_1 = \text{lbl}_2$ **and** $\text{dist}_{\text{ctr}} < \text{dist}_{\text{thresh}}$ **then**
    return True
  else
    return False
  end if
end procedure

Une fois générée, la couche des objets (illustrée sur la figure 9) offre une représentation claire et informative des objets détectés dans l'environnement. Chaque nœud du graphe est enrichi d'attributs tels que l'étiquette, la position et les dimensions. Ce positionnement précis et cette cartographie détaillée ouvrent la voie à des stratégies de navigation avancées.

2.4.3.3 Processus de cartographie des objets

Après la détection par le modèle Omni3D, chaque objet ω est initialisé dans le repère de la caméra par sa $bbox3D$. La transition vers les coordonnées de la carte implique une série de transformations pour aligner les objets détectés avec le repère global utilisé dans la carte. Ce processus de transformation est crucial pour positionner avec précision les objets dans l'environnement, en tenant compte de l'orientation du robot et de la perspective de la caméra. Chaque ω représente un objet détecté exprimé dans le repère de la caméra, tandis que μ correspond à la position de cet objet après transformation dans le repère global de la scène. Chaque objet cartographié μ dans l'espace global est défini comme :

$$\mu = (l, P, D, Q) \tag{2.2}$$

où :

— $P = (x', y', z')$ indique la position transformée de l'objet dans le repère global.

— $Q = (q_x, q_y, q_z, q_w)$ représente le quaternion décrivant l'orientation de l'objet par rapport au repère global ; il complète ainsi l'information de position du robot.

En adoptant cette formalisation, la couche des objets devient une collection structurée de tuples μ, chacun offrant une représentation détaillée de l'objet dans l'environnement. Cette structuration permet de mapper les objets en prenant en compte le référentiel de la carte métrique, puisque leur position est estimée relativement à la localisation du robot dans l'espace. En intégrant ces informations, la couche des objets assure une cohérence spatiale entre les objets détectés et la carte métrique sous-jacente.

2.4.4 Organisation spatiale et cartographie relationnelle

Pour qu'un robot puisse naviguer de manière autonome et efficace en utilisant les informations sémantiques de haut niveau, il doit être capable d'interpréter non seulement la disposition physique de l'espace, mais aussi les relations fonctionnelles entre les différentes zones et les objets présents. C'est dans cette optique que l'organisation spatiale et la cartographie relationnelle sont essentielles. Ces processus permettent de structurer l'environnement en entités distinctes, facilitant ainsi la planification des trajets et l'interaction du robot avec son environnement.

L'objectif principal de ces étapes est de fournir une représentation topologique enrichie de sémantique, où chaque espace est non seulement défini par ses frontières physiques, mais aussi par son contenu et ses relations avec d'autres zones. Cette structuration améliore l'intelligibilité de la carte pour le robot, qui peut ainsi prendre des décisions basées sur des concepts plus abstraits plutôt que sur de simples coordonnées spatiales.

2.4.4.1 Segmentation des pièces

L'un des premiers défis de l'organisation spatiale consiste à diviser l'environnement en zones cohérentes appelées pièces. Cette segmentation repose sur des caractéristiques structurelles et

sémantiques, utilisées pour identifier des espaces distincts dans l'environnement.

La segmentation des pièces est réalisée en exploitant plusieurs indices issus de la carte métrique et des objets présents :

— Indices géométriques : La segmentation repose sur des propriétés structurelles de l'espace, telles que la présence de surfaces fermées (murs), d'ouvertures localisées (portes) ou de passages restreints. Ces éléments définissent des discontinuités spatiales exploitables pour délimiter des zones fonctionnelles distinctes. L'analyse de la connectivité spatiale, de l'alignement des parois et des interruptions dans la carte d'occupation permet ainsi d'identifier les frontières potentielles entre pièces.

— Indices sémantiques : La densité et la distribution des objets dans l'espace peuvent fournir des indices complémentaires. Une forte concentration d'objets dans une zone peut indiquer une fonction particulière (comme une cuisine ou un bureau) et justifier une séparation sémantique d'autres zones plus dégagées.

Dans IntelliMap, la segmentation des pièces est réalisée en combinant plusieurs approches en s'appuyant sur des méthodologies existantes, telles que celles décrites dans [37], pour évaluer la possibilité de passer dans des zones et établir des limites claires entre les pièces. Cette étape est appliquée sur une carte de grille d'occupation de base de la HSTM.

Dans IntelliMap, la segmentation des pièces à partir de la grille d'occupation permet de structurer l'environnement en zones spatiales cohérentes correspondant aux différentes pièces du bâtiment. Chaque pièce segmentée est représentée par un nœud dans la couche des pièces de la HSTM, tandis que les arêtes du graphe correspondant aux arêtes entre pièces (room-to-room edges), modélisent les transitions navigables entre ces pièces. Par ailleurs, chaque zone segmentée est convertie en un polygone 2D représentant les limites géométriques de la pièce, puis stockée dans la structure hiérarchique de la HSTM. Chaque pièce est représentée par un polygone 2D $\mathcal{P}_i$ dans le repère global, défini par un ensemble de sommets $\{p_1, p_2, ..., p_n\}$.

Bien que la méthode soit automatisée, des ajustements manuels peuvent être nécessaires pour corriger certaines erreurs, comme la fusion de couloirs longs subdivisés en plusieurs pièces.

La flexibilité du framework permet également l'intégration d'autres méthodes de segmentation, garantissant une adaptabilité aux différents scénarios et besoins. L'essentiel est de produire une liste de nœuds (pièces) avec leurs limites représentées par des polygones 2D et les connectivités entre elles. Grâce à la flexibilité d'IntelliMap, d'autres méthodes de segmentation peuvent être intégrées, qu'elles reposent sur une grille d'occupation, un nuage de points 3D, ou encore des modèles sémantiques avancés. Le choix de la méthode dépend du type de données disponible et du niveau de granularité ou de sémantique requis par l'application robotique visée.

2.4.4.2 Catégorisation des lieux

Après la segmentation, il est crucial d'assigner un label sémantique à chaque pièce, afin d'enrichir la compréhension du robot. Cette catégorisation permet au robot d'interpréter les espaces suivant leur fonction et les objets caractéristiques qu'ils contiennent.

Un Large Language Model (LLM) ou une ontologie peuvent être utilisés pour attribuer des étiquettes sémantiques à chaque pièce. La catégorisation des lieux est essentielle pour plusieurs raisons. Elle améliore la planification des trajets : un robot à la recherche d'un objet spécifique peut restreindre son exploration aux pièces les plus susceptibles de le contenir. Elle facilite également l'interaction homme-robot, en permettant au système d'interpréter des commandes plus naturelles, telles que « Va dans la cuisine », grâce à l'utilisation d'une carte annotée sémantiquement. En structurant l'environnement par catégories fonctionnelles, elle réduit l'ambiguïté en navigation et facilite l'élaboration de stratégies de déplacement adaptées.

Dans le cadre d'IntelliMap, la catégorisation des pièces est réalisée à l'aide de modèles de LLM, tels que GPT-4o, qui analysent les objets présents et la disposition spatiale pour attribuer une catégorie pertinente à chaque pièce. Cette approche permet d'automatiser l'annotation sémantique en exploitant les vastes connaissances des modèles entraînés sur des données variées. Chaque pièce est ainsi classifiée dans une catégorie préétablie (ex. "cuisine", "bureau", "chambre", etc.).

Pour chaque pièce détectée après segmentation, une liste d'objets contenus dans cet espace est extraite. Cette liste est ensuite utilisée comme entrée pour un LLM, avec un prompt structuré spécifiant les catégories possibles et demandant une classification unique. Le prompt utilisé est le suivant :

```
Task: Identify the type of room based on the objects it contains.
Choose only one room type from the list below.

Room types: [living room, dining room, kitchen, bedroom,
bathroom, home office, laundry room, hallway, entryway]
Objects in the room: [list of detected objects]

Respond with only the room type in this format:
room type
```

Le modèle retourne alors une classification unique pour chaque pièce comme dans l'exemple présenté Figure 11. Cette méthode a été appliquée sur plusieurs pièces, et les résultats obtenus illustrée dans le tableau 2.2 ont été comparés aux pièces de référence. Par exemple avec GPT-4o, environ 90% des classifications générées correspondent à la catégorisation de référence.

Certaines erreurs de classification peuvent néanmoins survenir, notamment lorsque des objets de catégories fonctionnelles différentes coexistent dans une même pièce (par exemple, la présence simultanée d'un bureau et d'un micro-ondes). Ces situations ambigües peuvent conduire à une interprétation erronée de la fonction dominante de la pièce. La méthodologie peut cependant être ajustée selon les besoins spécifiques de l'application, le contexte, et complétée par des corrections manuelles lorsque la classification automatique est jugée incertaine ou incohérente.

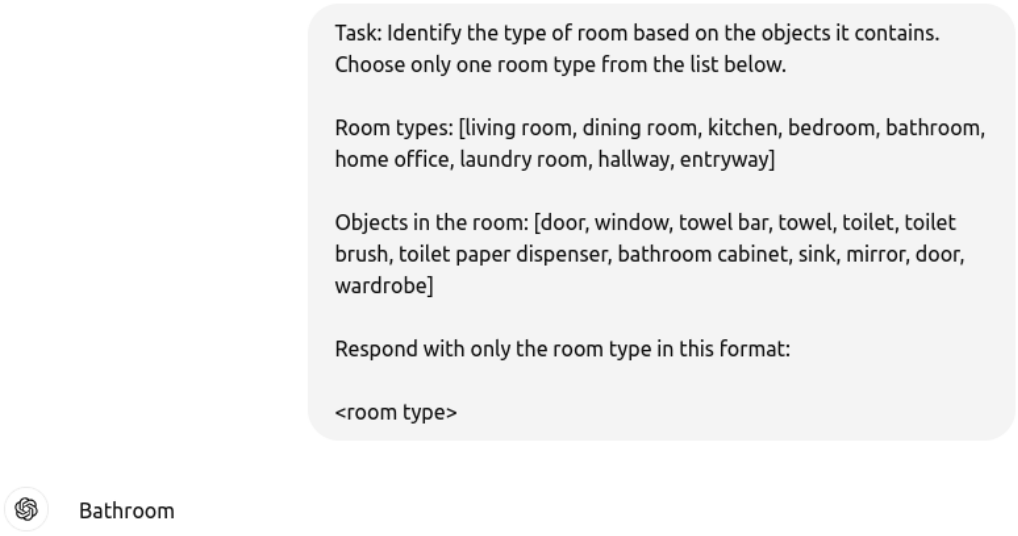


FIGURE 11 – Exemple de catégorisation d'une pièce avec GPT-4o

2.4.4.3 Processus de génération des arêtes

— Arêtes entre pièces (room-to-room edges) : Les arêtes entre pièces sont générées lors du processus de segmentation des pièces, où chaque grille d'occupation 2D (correspondant à un étage) est analysée indépendamment pour structurer l'environnement en pièces distinctes. Sur chaque étage, les pièces sont identifiées selon leur connectivité géométrique. Ces arêtes sont intégrées au graphe de sortie, illustrant la traversabilité entre les nœuds représentant les pièces. Une connexion est établie entre deux pièces lorsque leur accessibilité est détectée.

Les transitions entre pièces situées à des étages différents (par exemple via un escalier) doivent être modélisées explicitement dans la carte sémantique. Actuellement, IntelliNav prend en charge ces connexions verticales uniquement dans le cas de transitions par escaliers, en exploitant des waypoints inter-étages pré-calculés, extraits au cours de la phase de cartographie. Lorsqu'un robot navigue entre étages (montée ou descente d'un escalier), sa trajectoire réelle est enregistrée via un processus de SLAM, puis segmentée en une séquence de waypoints directionnels reliant une pièce source à une pièce destination.

Chaque arête inter-étages enregistrée dans la HSTM inclut deux listes de waypoints 3D : une liste des points intermédiaires pour passer de la pièce r_i (étage f_i) à la pièce r_j (étage $fi + 1$) et la liste inverse pour effectuer la descente.

Ces chemins de waypoints sont définis par des coordonnées tridimensionnelles (x, y, z), et peuvent être enrichis par des paramètres supplémentaires (par exemple des consignes de vitesse) pour faciliter le contrôle bas-niveau fluide du robot pendant les montées ou descentes.

— Arêtes entre pièces et objets (room-to-object edges) : Pour établir les arêtes entre les pièces et les objets dans la structure de la carte, le processus s'appuie sur les informations stockées dans le fichier JSON contenant les nœuds des pièces et des objets. Chaque

pièce est représentée par un polygone 2D $\mathcal{P}_i$ et chaque objet o_j est décrit par une boîte englobante tridimensionnelle $bbox3D_j$, définie par son centroïde $C_j = (x_j, y_j, z_j)$ et ses dimensions. La couche des objets étant déjà enregistrée dans le fichier JSON de la carte, les coordonnées de chaque objet sont référencées dans le même repère global que les pièces. A partir de ces informations, un processus de vérification d'inclusion est appliqué afin de déterminer les relations spatiales et d'enregistrer les arêtes correspondantes dans la structure HSTM :

— Pour chaque objet o_j, on vérifie si C_j appartient à une pièce P_i en testant son inclusion dans le polygone $\mathcal{P}_i$. Cette vérification est réalisée via un test de point dans un polygone 2D.

— Si $C_j \in \mathcal{P}_i$, une arête est créée et enregistrée dans la structure JSON.

Cette arête établit la relation spatiale entre la pièce et l'objet, indiquant que l'objet appartient à cette pièce.

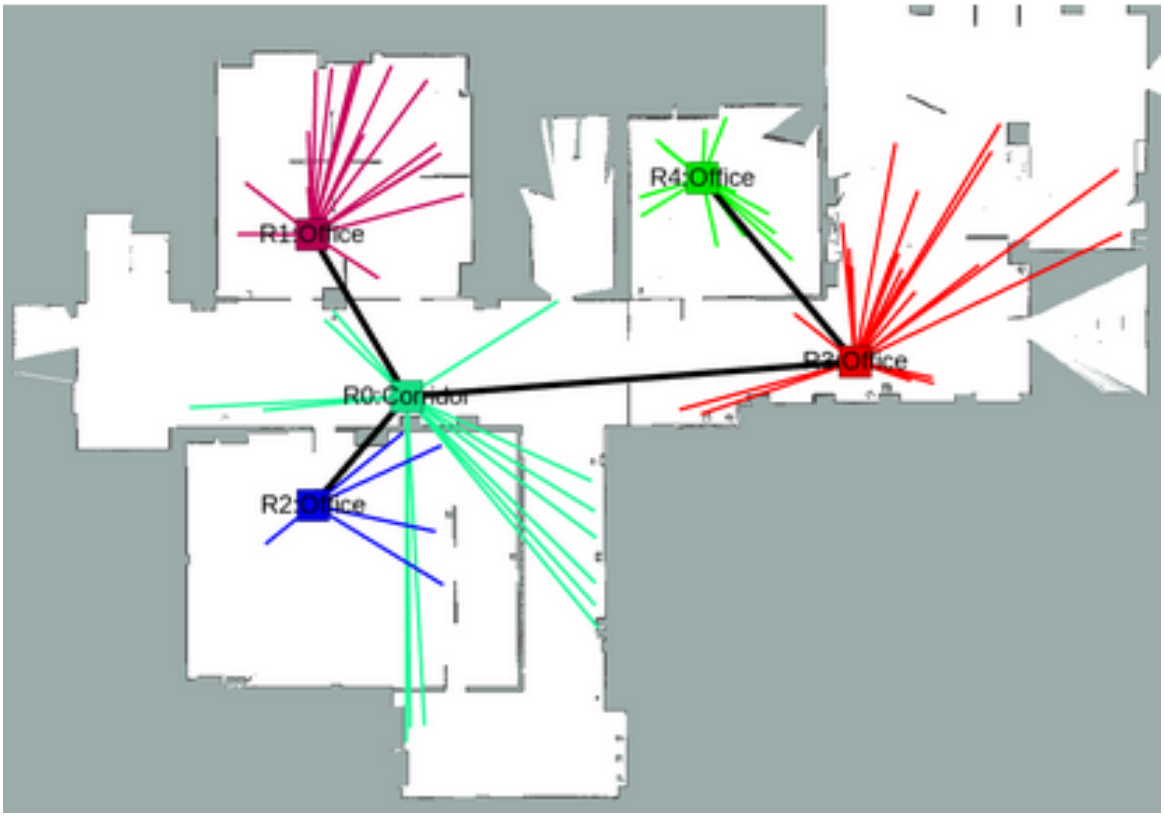


FIGURE 12 – Cartographie des pièces

Comme illustré dans la figure 12, la segmentation des pièces permet d'obtenir une carte topologique où chaque pièce est identifiée et connectée aux autres selon leur accessibilité. Les objets sont aussi associés à leurs pièces respectives.

2.4.4.4 Segmentation des bâtiments

La segmentation des bâtiments dans IntelliMap structure l'environnement à un niveau plus global en distinguant des entités architecturales distinctes. Actuellement, cette segmentation

est réalisée manuellement, car dans la majorité des environnements industriels et tertiaires, le nombre de bâtiments annexes est limité et leur distinction est explicite. Cette approche manuelle présente plusieurs avantages. Elle garantit une identification et un étiquetage corrects des bâtiments, en particulier dans des environnements complexes où les distinctions architecturales peuvent être subtiles. Dans des environnements à petite ou moyenne échelle, une segmentation manuelle ne représente pas une charge de travail excessive. De plus, la relation spatiale entre les pièces, essentielle pour la navigation, est déjà capturée à travers la segmentation des pièces et le graphe topologique des connexions traversables. Cela garantit une intégration cohérente de la segmentation des bâtiments.

À l'avenir, cette segmentation pourrait être automatisée en s'appuyant sur des plans architecturaux numérisés ou des données LiDAR.

2.5 Conclusion

Le framework IntelliMap propose une approche structurée pour la génération de cartes sémantiques hiérarchiques, essentielles à la navigation autonome et à l'interaction contextuelle des robots dans des environnements complexes. En combinant des informations métriques, sémantiques et topologiques, IntelliMap offre une représentation riche et intuitive de l'environnement, favorisant une compréhension et une interaction efficaces du robot avec son espace de travail. En combinant des techniques en SLAM, en détection d'objets et en segmentation spatiale, ce framework offre une solution complète et adaptable pour la modélisation d'environnements complexes. La structure modulaire d'IntelliMap facilite l'intégration de nouvelles méthodes et technologies, garantissant une évolution continue du framework. Les travaux futurs pourraient inclure l'amélioration des algorithmes de segmentation, l'intégration de modèles sémantiques plus sophistiqués et l'optimisation des performances pour des environnements à grande échelle. La HSTM constitue la carte de base sur laquelle s'appuie la navigation sémantique avec le framework IntelliNav. Dans le présent chapitre, une HSTM à trois couches a été définie : la couche métrique, la couche des objets et la couche des pièces. Cette structure n'est pas figée et ne constitue pas un modèle universel. Elle a été spécifiquement choisie pour valider le fonctionnement du système de navigation proposé dans un environnement mono-bâtiment, en permettant une articulation claire entre la géométrie de l'espace, les objets détectés et les pièces fonctionnelles.

Cette structuration à trois couches repose sur une représentation construite hors ligne, ce qui est cohérent avec la vocation de la HSTM à modéliser des connaissances sémantiques relativement stables sur l'environnement. La génération de la carte n'est pas entièrement automatisée, mais repose sur un processus modulaire rendu possible par la flexibilité d'IntelliMap. Chaque couche (métrique, des objets, des pièces) peut être générée ou régénérée indépendamment, en fonction des données disponibles et des modèles employés. Cela permet d'adapter dynamiquement la structure de la carte aux besoins spécifiques de l'application. Cette approche est particulièrement pertinente dans un contexte industriel, où les environnements, les contraintes et

les objectifs de navigation peuvent varier d'un projet à l'autre. Elle s'inscrit dans le cadre d'une collaboration université–entreprise visant à concilier recherche avancée et besoins concrets du terrain. Le chapitre suivant exploitera cette représentation pour introduire les mécanismes de navigation sémantique dynamique dans des environnements partiellement connus ou évolutifs.

ID Pièce	Objets	Nom Réf.	GPT-4o	o1-Preview	o1-Mini
3009	door, window, towel bar, towel, toilet, toilet brush, toilet paper dispenser, bathroom cabinet, sink, mirror, door, wardrobe	bathroom	bathroom	bathroom	bathroom
3008	door, toilet, bathroom cabinet, sink, air conditioner, bathtub, shower curtain, window, trash can, mirror	bathroom	bathroom	bathroom	bathroom
3007	decoration, window, painting, bed, lamp, cabinet, book, decoration, window, pillow, pillow, pillow	bedroom	bedroom	bedroom	bedroom
3006	door, door, window, bed, pillow, pillow, pillow, lamp, table, decoration, shelf	bedroom	bedroom	bedroom	bedroom
3005	carpet, door, bed, rack, lamp, table, book, wall panel, painting, window, pillow, pillow, pillow	bedroom	bedroom	bedroom	bedroom
3004	door	hallway	hallway	entryway	entryway
3003	door, painting,couch, table, rug, stool, refrigerator cabinet, refrigerator, cabinet, sink, decoration, vase, decoration, table, flower vase, tray, decoration, flowerpot, flower vase, shelf, window, bench, pillow, bowl, stove, microwave, oven, towel, decoration, dishwasher, picture, table	kitchen	kitchen	kitchen	kitchen
3002	carpet, door, wall panel, window, decoration, wall lamp, door, mirror, mirror, decoration, chair, chair, desk, desk chair, flowerpot, book	living room	home office	home office	home office
3001	door, painting, window, flowerpot, podium, wall lamp, dining table, flower vase, dining chair, dining chair, plate, plate	dining room	dining room	dining room	dining room
3000	rug, couch, sofa chair, sofa chair, fireplace, shelf, podium, decoration, flower vase, flower vase, flowerpot, book, picture, lamp, wall panel, decoration, book, table, podium, step, wall lamp, door, pillow, blanket, window, mirror	living room	living room	living room	living room

TABLE 2.2 – Résultats de catégorisation des lieux : comparaison entre les sorties ChatGPT et les types de pièces de référence.

CHAPITRE 3

Navigation Sémantique avec IntelliNav

3.1 Introduction

La navigation sémantique permet aux robots de raisonner sur leur environnement au-delà des simples informations géométriques. Contrairement aux approches traditionnelles, qui se concentrent principalement sur la détection d'obstacles et la planification de trajectoires basées sur des cartes statiques, la navigation sémantique intègre une compréhension contextuelle de l'espace. Cette capacité est particulièrement cruciale dans des environnements dynamiques, où des objets peuvent être déplacés, des chemins bloqués ou des configurations modifiées de manière imprévisible. Dans de tels scénarios, les méthodes classiques de navigation, souvent limitées par leur dépendance à des cartes préétablies et à des algorithmes de planification rigides, peinent à s'adapter efficacement.

L'intégration d'informations sémantiques dans les systèmes de navigation robotique est devenue un enjeu clé pour améliorer l'autonomie et l'efficacité des robots dans des environnements complexes. Par exemple, dans le domaine de la robotique d'assistance, comme le démontre le projet ENRICHME [38], la navigation sémantique joue un rôle essentiel pour permettre aux robots d'interagir de manière fiable avec des utilisateurs vulnérables, en surveillant leur santé ou en facilitant les interactions sociales. De même, dans les environnements industriels, la capacité à interpréter les espaces fonctionnels et à s'adapter aux changements fréquents, tels que la réorganisation des espaces de travail ou le déplacement d'objets, est cruciale pour assurer des opérations fiables et efficaces.

IntelliNav a été conçu pour répondre à ces défis en exploitant pleinement les cartes sémantiques générées par IntelliMap, présenté dans le chapitre précédent. IntelliNav permet une navigation robotique adaptative, efficace et explicable dans des environnements intérieurs dynamiques, en s'appuyant sur une carte HSTM à trois couches (métrique, des objets et des

pièces).

IntelliNav combine ces cartes prédéfinies avec des mises à jour dynamiques et une replanification au niveau des trajectoires. IntelliNav permet une planification globale de chemin sémantique, tout en intégrant un mécanisme de mise à jour dynamique de la carte pour refléter les changements environnementaux en temps réel. De plus, le framework intègre un module de mémoire persistante au sein de la HSTM, enregistrant les difficultés de navigation rencontrées pour permettre au robot d'éviter les obstacles précédemment identifiés ou les zones difficiles à traverser et aussi les succès de navigation afin de permettre au robot d'optimiser les trajets en fonction des zones validées comme accessibles. Cette combinaison de planification sémantique, de mises à jour dynamiques et de mémoire persistante constitue un système cohérent, fiable et efficace pour la navigation en environnements intérieurs dynamiques.

Ce chapitre est structuré comme suit. La section 3.2 présente un état de l'art des approches existantes en matière de navigation sémantique dynamique. La section 3.3 introduit l'architecture globale du framework IntelliNav et ses principales fonctionnalités. Enfin, la section 3.4 propose une synthèse des contributions présentées dans ce chapitre.

3.2 État de l'art

La navigation sémantique nécessite des stratégies intelligentes. En s'appuyant sur des représentations sémantiques de l'environnement, un robot peut non seulement percevoir son environnement, mais aussi raisonner sur ses propriétés et anticiper des modifications potentielles à différentes échelles temporelles, à l'image des humains qui exploitent leur compréhension sémantique pour prédire l'évolution de leur environnement [39]. Cependant, plusieurs défis persistent et limitent encore l'efficacité des systèmes de navigation sémantique existants.

Naviguer et comprendre des environnements complexes sur de longues périodes représente un défi majeur pour les robots. Lors de leur déploiement, ceux-ci perçoivent un large éventail d'objets, d'événements dynamiques et de phénomènes difficiles à encapsuler dans des représentations conventionnelles telles que les cartes métriques ou sémantiques. Ces limitations compliquent l'adaptation aux environnements en évolution constante et restreint la capacité des robots à exploiter leur historique d'interactions pour améliorer leur navigation [40].

La représentation de l'environnement doit non seulement intégrer des informations sémantiques, mais aussi permettre une mise à jour efficace face aux changements rencontrés. Au-delà de la mise à jour de la carte en temps réel, un enjeu fondamental réside dans la gestion de la mémoire sur le long terme. Les robots déployés sur de longues durées devraient être capables de raisonner sur l'historique de leurs interactions. Une telle capacité nécessite un mécanisme de récupération et d'exploitation efficace des expériences passées, afin d'enrichir la prise de décision et de favoriser une navigation adaptative et explicable [40]. La planification de trajectoire doit rester suffisamment flexible pour s'adapter aux données nouvellement perçues, tout en capitalisant sur l'historique des navigations précédentes. En intégrant les retours d'expérience mémorisés, le robot peut ajuster ses parcours de manière proactive, optimisant ainsi ses

déplacements sans compromettre leur efficacité.

Dans ce contexte, nous introduisons IntelliNav, un framework de navigation sémantique conçu pour répondre aux exigences des environnements dynamiques. Cette section propose une analyse approfondie des travaux existants en navigation sémantique, en mettant en lumière les méthodes de planification de trajectoire et les stratégies de mise à jour des cartes et de la replanification. Cette revue situe IntelliNav dans l'état de l'art et met en évidence la manière dont son approche répond aux limitations des systèmes actuels.

3.2.1 Planification statique vs planification adaptative

La planification de trajectoire est un élément central de la navigation sémantique du robot. Elle consiste à déterminer un chemin optimal pour atteindre un objectif tout en tenant compte des contraintes de l'environnement et en intégrant des informations sémantiques sur l'environnement. Les approches de planification en navigation sémantique peuvent être classées en deux grandes catégories : la planification statique et la planification adaptative. Ces deux méthodes se distinguent principalement par leur capacité à s'adapter aux changements environnementaux et par la manière dont elles exploitent l'information sémantique pour ajuster les trajectoires.

3.2.1.1 Planification statique

La planification statique dans le cadre de la navigation sémantique repose sur l'intégration d'informations sémantiques dans la planification de trajectoire, en s'appuyant sur des cartes prédéfinies et des algorithmes de recherche de chemin classiques, comme illustré dans plusieurs travaux [23, 41, 42, 43]. Cette approche consiste à calculer un chemin global optimal avant l'exécution de la tâche de navigation, en supposant que l'environnement est stable et parfaitement connu.

Cette planification est généralement réalisée sur une représentation topologique ou directement sur une carte métrique enrichie d'informations sémantiques. Le chemin est calculé en fonction d'une carte préétablie et suit une trajectoire optimisée vers l'objectif. Lors de l'exécution de la trajectoire, de légères modifications locales peuvent être effectuées, qui se basent sur une carte métrique pour contourner des obstacles temporaires tant que le passage vers l'objectif reste accessible. Toutefois, si des obstacles persistants, comme une porte fermée ou un passage obstrué, empêchent la progression, la planification statique atteint rapidement ses limites et nécessite une réévaluation complète de la trajectoire.

Dans une approche classique, la planification statique d'une trajectoire globale s'appuie sur des graphes topologiques, où les nœuds représentent des emplacements significatifs (pièces, zones de passage) et les arêtes définissent les transitions entre ces espaces. La navigation est alors réalisée en suivant les relations entre espaces, souvent à un niveau abstrait, sans prendre directement en compte la géométrie détaillée de l'environnement. Des algorithmes de recherche de chemin, tels que A* (A-star) [44], Dijkstra [45] ou RRT (Rapidly-exploring Random Tree) [46], sont couramment utilisés pour optimiser les trajectoires en fonction de critères comme la

distance, le temps de parcours ou la complexité des transitions entre ces espaces. [47] fournit une analyse approfondie d'algorithmes de planification, mettant en lumière leurs avantages et leurs limites.

Par exemple, Le SMaNa [41] propose une approche de navigation sémantique basée sur un graphe 2.5D construit en ligne à partir de données de cartographie tridimensionnelle. Ce graphe est généré à partir de Semantic Octomap [1], ou de Kimera-TSDF [2], où chaque voxel est associé à une classe sémantique, une hauteur et des coordonnées absolues, facilitant ainsi la structuration de l'environnement. La navigation repose sur un algorithme A* pondéré, intégrant des coefficients de traversabilité liés aux classes sémantiques des environnements. Toutefois, cette approche repose sur une carte maintenue statiquement une fois construite. Elle ne prend pas en compte les évolutions dynamiques de l'environnement au cours de la navigation, ce qui limite sa capacité d'adaptation dans des contextes changeants.

S-Nav [23] propose une planification hiérarchique sémantico-géométrique, combinant un graphe sémantique de haut niveau structuré par pièces et portes avec une planification géométrique. La planification sémantique, une fois effectuée, ne se réadapte pas en continu, ce qui signifie que le chemin global reste déterminé de manière statique après le premier calcul. L'intégration d'un planificateur géométrique permet une meilleure flexibilité locale, mais la navigation reste contrainte par la structure initiale du graphe sémantique.

Une autre approche consiste à générer directement le chemin sur une carte métrique enrichie de données sémantiques, où les informations sur les objets, les espaces ou les relations entre éléments sont directement intégrées à la carte géométrique. Dans ce cas, la planification est réalisée en prenant en compte à la fois les coûts métriques et les informations sémantiques. Dans ce cadre, [42] propose une structure multi-couches intégrant une grille de traversabilité dérivée d'un nuage de points sémantique, ainsi que des contraintes liées à l'exploration et à l'observation d'objets spécifiques. Un algorithme A* modifié et une version adaptée du T-RRT sont utilisés pour optimiser les trajets tout en respectant les contraintes de la carte sémantique. Cependant, cette approche reste statique, car les trajectoires sont planifiées sur une carte figée et nécessitent un recalcul complet en cas de modification majeure de l'environnement. [43] propose une approche basée sur une carte sémantique multi-couches enrichie et un Bi-RRT [3]. Cette méthode intègre une gestion des objets lors de la navigation, en distinguant les obstacles permanents des objets mobiles. Lorsqu'un objet mobile bloque temporairement le passage, le robot peut adapter localement sa trajectoire en le déplaçant vers une zone appropriée. Toutefois, en cas d'obstacle persistant ou de modification structurelle, aucune replanification globale n'est effectuée, ce qui limite l'adaptabilité de l'approche face à des environnements dynamiques.

1. Semantic OctoMap est une extension sémantique de l'OctoMap, une structure de données hiérarchique utilisée pour représenter l'occupation de l'espace en 3D sous forme de voxels. Chaque voxel est enrichi par une étiquette sémantique, en plus de l'information d'occupation.

2. Kimera-TSDF est une méthode de reconstruction dense en 3D basée sur un champ de distance signé tronqué (TSDF), intégrant à chaque voxel une classe sémantique ainsi qu'une estimation de surface, permettant une représentation fine de l'environnement.

3. Bi-RRT (Bi-directional Rapidly-Exploring Random Tree) est une variante de l'algorithme RRT qui construit simultanément deux arbres à partir du point de départ et du but.

L'une des principales faiblesses de cette approche réside dans son incapacité à s'adapter aux changements imprévus de l'environnement en temps réel. Lorsqu'un obstacle persistant apparaît, comme une porte fermée ou un meuble déplacé bloquant un passage clé, la planification statique ne permet pas d'ajustement de la trajectoire et compromet la fluidité de la navigation. Dans un contexte où la navigation sémantique vise une interaction plus naturelle avec l'environnement et les humains, la planification statique peut se révèler insuffisante.

3.2.1.2 Planification adaptative

La planification adaptative permet au robot de replanifier globalement sa trajectoire en cas de problème majeur, comme l'apparition d'un obstacle imprévu ou un changement dans l'objectif. Contrairement à la planification statique, où le chemin global est fixe, la planification adaptative recalcule tout ou partie de la trajectoire initiale pour s'adapter à la nouvelle situation. Des algorithmes de planification de chemins sont utilisés pour ajuster les trajectoires. La planification adaptative offre une meilleure réactivité dans des environnements dynamiques, en assurant l'adaptation du robot face aux changements imprévus. Cependant, cette approche peut présenter des défis, notamment une charge computationnelle plus élevée due aux recalculs globaux, en fonction de la complexité de la représentation sémantique de l'environnement. Dans ce cadre, [48] propose une approche hiérarchique combinant une planification sémantique globale et une planification géométrique locale, tout en intégrant des contraintes sociales. L'architecture, CORTEX, permet d'exploiter un graphe sémantique enrichi d'informations sociales, où les objets et les pièces sont connectés à l'aide de relations symboliques et géométriques. Cette structure facilite la prise en compte de la présence humaine, en ajustant les trajectoires pour respecter les distances sociales et les intentions d'interaction.

Il est important de noter que la navigation locale (qui concerne les ajustements mineurs pour éviter des obstacles ou suivre une trajectoire préétablie) est distincte de la planification adaptative. Alors que la planification adaptative recalcule globalement la trajectoire, la navigation locale se concentre sur des ajustements à court terme, comme le suivi de trajectoire ou l'évitement d'obstacles à l'aide de capteurs locaux (par exemple, des capteurs LiDAR ou des caméras). Ces deux approches sont souvent combinées pour offrir une navigation robuste et flexible.

Les planifications statique et adaptative représentent deux approches complémentaires, chacune ayant ses forces et ses faiblesses. Alors que la planification statique est efficace dans des environnements contrôlés et prévisibles, la planification adaptative est indispensable pour naviguer dans des environnements dynamiques et partiellement connus. Cependant, les méthodes existantes souffrent souvent d'un manque d'adaptabilité aux changements imprévus ou d'une charge computationnelle élevée. C'est dans ce contexte que IntelliNav propose une solution innovante, combinant les avantages des deux approches. Grâce à la HSTM, IntelliNav permet une planification hybride, où des trajectoires globales prédéfinies sont ajustées en temps réel en fonction des informations perçues. Cette approche offre une navigation robuste, adaptative et

explicable. Après avoir exploré les méthodes de planification, nous nous intéressons maintenant à des approches plus récentes, comme la navigation guidée par le langage naturel, qui exploitent des modèles de langage pour améliorer l'interaction entre le robot et son environnement.

3.2.2 Navigation guidée par le langage naturel et les LLM

Récemment, l'intégration des modèles de LLM a ouvert de nouvelles perspectives pour la navigation sémantique, en favorisant une interaction plus intuitive entre le robot et son environnement. Ces approches [49, 50] exploitent des descriptions en langage naturel et des représentations sémantiques structurées, telles que les graphes de scène 3D, pour guider la planification et l'exécution des tâches de navigation. Les approches basées sur les LLMs permettent une compréhension flexible des commandes utilisateur. Par exemple, un robot peut interpréter des instructions en langage naturel, comme "Va chercher la tasse sur la table de la cuisine", et les transformer en actions exécutables. [51] propose une approche combinant les 3DSGs avec les LLM afin de structurer un processus de navigation en trois étapes : la construction d'un 3DSG hiérarchique, enrichi avec des descriptions sémantiques pour structurer un espace navigable ; l'exploitation des LLMs pour effectuer une recherche sémantique au sein du graphe et identifier les sous-structures pertinentes pour accomplir une tâche de navigation ; la génération d'un plan d'actions intégrant des instructions de déplacement et de manipulation. [52] propose MoMa-LLM, une approche qui associe les LLMs à des graphes de scène 3D dynamiques, afin que le robot puisse exécuter des tâches de navigation et de manipulation en exploitant une compréhension sémantique enrichie de son environnement. RoboHop [53] propose une approche de navigation sémantique qui repose sur une représentation topologique de l'environnement, où les nœuds du graphe correspondent à des segments d'image sémantiquement pertinents. L'intégration d'un LLM permet d'effectuer des requêtes en langage naturel, facilitant ainsi la navigation basée sur des descriptions d'objets et de relations spatiales. Par exemple, un utilisateur peut spécifier un objectif sous la forme "Va vers la chaise à côté de la fenêtre", et le système interprète cette instruction pour générer une trajectoire navigable. Un autre axe de recherche consiste à traduire les instructions en langage naturel en une représentation logique exploitable pour la navigation. Par exemple, une approche récente [54] utilise un modèle LLM pour convertir une instruction en logique temporelle linéaire (LTL), qui est ensuite utilisée pour générer un plan de navigation optimal sur un graphe de scène hiérarchique.

Cependant, bien que ces méthodes aient démontré des améliorations en matière d'interprétation sémantique et d'adaptabilité aux environnements complexes, elles s'appuient fortement sur des modèles génératifs, ce qui peut introduire des incertitudes et des erreurs d'interprétation susceptibles d'affecter la robustesse de la navigation. Un autre défi est la dépendance aux prompts et aux exemples. La performance des LLMs dépend largement de la qualité du prompt fourni, ce qui introduit une variabilité difficile à contrôler. Par exemple, si une instruction est ambiguë ou mal formulée, le modèle de langage peut générer un plan incorrect ou incomplet. Cette dépendance aux prompts nécessite une attention particulière lors de la conception des

interfaces utilisateur et des systèmes de dialogue pour les robots. [51] introduit des techniques d'ingénierie de prompt et des mécanismes de notation des réponses, dans le but de réduire l'incertitude des modèles génératifs et d'améliorer la robustesse de la planification. [55] utilise un LLM comme heuristique pour orienter la recherche de chemin. Cette approche combine des prompts positifs et négatifs pour améliorer la robustesse des scores attribués aux sous-objectifs et emploie une stratégie de prompting pour affiner le raisonnement spatial.

Enfin, certaines de ces méthodes, par exemple [51, 53], ne stockent pas une mémoire des navigations passées, ce qui signifie que chaque navigation est traitée de manière indépendante. Le robot ne peut pas tirer parti des expériences passées pour améliorer ses performances, et les erreurs peuvent être répétées. Par exemple, si un obstacle est rencontré dans une mission précédente, le robot peut le redécouvrir plusieurs fois sans que cela affecte la planification future.

3.2.3 Limites des approches traditionnelles en navigation sémantique

Les premiers approches traditionnelles de navigation sémantique se concentrent sur l'intégration d'informations sémantiques dans la planification de trajectoires, en s'appuyant sur des cartes prédéfinies et des algorithmes de recherche de chemin classiques, [23, 41, 42, 43]. Ces méthodes offrent des capacités de navigation de base, mais elles sont souvent limitées par la richesse sémantique des cartes utilisées, qui ne contiennent généralement que des informations fondamentales telles que les emplacements des objets, les portes ou les étiquettes de pièces. Par exemple, S-Nav [23] propose une architecture hiérarchique qui améliore la planification géométrique en exploitant des informations sémantiques sur les limites des pièces et les portes. Cependant, cette approche ne permet pas d'attribuer des étiquettes sémantiques spécifiques aux pièces, comme "cuisine" ou "bureau". De même, [41] propose une navigation basée sur un graphe sémantique 2.5D et utilise un algorithme A* pondéré pour planifier des trajectoires optimales. Une carte sémantique composée de trois couches est utilisée dans [42] : une costmap, une grille d'exploration conventionnelle et une grille binaire qui enregistre les observations de catégories sémantiques distinctes.

Bien que ces méthodes offrent des fonctionnalités de base, elles manquent souvent de capacité de mémoire et de replanification dynamique, limitant leur adaptabilité dans des environnements complexes et dynamiques. Ces approches présentent ainsi plusieurs limitations. Elles s'appuient sur des cartes statiques qui ne s'adaptent pas aux modifications dynamiques de l'environnement. Elles ne prennent pas en charge la mémoire persistante. Elles ne permettent pas de raisonner sur des structures plus complexes, comme la hiérarchie des relations entre objets et pièces et les informations sémantiques représentées sont limitées.

3.2.4 Navigation sémantique dynamique

L'adaptation aux environnements dynamiques est un défi central dans la navigation sémantique des robots. Des avancées récentes exploitent les représentations sémantiques et les modèles d'apprentissage automatique pour améliorer l'adaptabilité. Les LLMs sont devenus des outils puissants pour interpréter les instructions et guider les robots dans des environnements complexes et dynamiques.

— Navigation sans cartes prédéfinies : Les approches "Object Goal Navigation" (ObjectNav) permettent aux robots de rechercher un objet cible dans un environnement inconnu, sans disposer d'une carte préalable. Ces méthodes sont particulièrement utiles dans des scénarios où l'environnement est non structuré, dynamique ou partiellement observable. Contrairement aux approches traditionnelles qui s'appuient sur des cartes prédéfinies, l'ObjectNav repose sur une exploration active, où le robot construit progressivement une représentation de l'environnement tout en cherchant l'objet cible. Dans ces méthodes, le robot progresse de manière exploratoire, en s'appuyant sur une combinaison d'indices visuels, de relations objets-pièces et parfois d'inférences sémantiques pour optimiser sa recherche.

Certaines approches [24, 56, 57, 58, 59, 53], se concentrent sur la construction incrémentale de cartes sémantiques pendant l'exploration. Ces méthodes permettent aux robots de localiser des objets spécifiques dans des environnements inconnus. Ces méthodes exploitent les avancées en vision par ordinateur et en apprentissage profond pour associer les observations du robot à des structures sémantiques exploitables par des algorithmes de navigation.

Cependant, ces approches présentent certaines limites en matière d'adaptabilité aux environnements dynamiques. En particulier, les mises à jour des cartes sémantiques sont souvent réactives et fonctionnent selon un mode « découverte », c'est-à-dire qu'elles intègrent principalement de nouvelles informations perçues pendant l'exploration, sans reconsidérer les éléments déjà cartographiés. Ainsi, si des objets sont déplacés ou disparaissent, la carte ne reflète plus fidèlement l'état actuel de l'environnement, ce qui peut allonger les temps de recherche et nuire à la robustesse du système.

Par exemple, RoboHop [53] présente une approche de cartographie topologique basée sur les segments. Cette méthode est limitée dans sa capacité à gérer les changements dynamiques de l'environnement. Dans ce travail, décrit dans [59], les mises à jour des arêtes n'impliquent pas la validation ou la correction des arêtes existantes ; elles se concentrent plutôt sur l'ajout de nouvelles arêtes au fur et à mesure que l'agent identifie des connexions entre des nœuds qui n'étaient pas liés auparavant.

— Approches de mise à jour dynamique des cartes : Les mises à jour dynamiques des cartes visent à garantir que la représentation de l'environnement par le robot reflète son état actuel, en intégrant des changements tels que l'ajout de nouveaux objets, la modification de la position des obstacles ou l'évolution des connexions entre les pièces.

Cette adaptabilité peut être obtenue soit en modifiant des cartes prédéfinies, soit en construisant progressivement de nouvelles cartes au cours de la navigation.

Alors que certaines méthodes utilisent des cartes statiques, d'autres intègrent des mises à jour par le biais d'interactions avec l'environnement, souvent facilitées par des outils tels qu'un LLM. Cependant, ces mises à jour sont généralement limitées à la mémoire épisodique, réinitialisée après chaque tâche ou épisode, et se concentrent principalement sur les objets avec lesquels le robot interagit directement. Les méthodes de mise à jour de cartes prédéfinies exploitent des structures existantes, telles que des graphes de scène ou des cartes sémantiques, qui sont mises à jour dynamiquement lorsque le robot rencontre de nouvelles informations. Ces mises à jour peuvent inclure l'ajout ou la suppression d'objets, la modification des connexions ou l'actualisation des propriétés en fonction des observations.

Par exemple, [60] propose une approche basée sur des graphes de scène, où les objets et leurs relations sont stockés sous forme de nœuds et d'arêtes. Cette représentation permet d'actualiser l'état du monde de manière progressive et structurée lors de l'exécution de tâches robotiques guidées par des instructions en langage naturel. À chaque interaction avec l'environnement, les graphes de scène et de robot sont mis à jour pour refléter les changements dus à l'exécution de la tâche. L'aspect clé de cette mise à jour réside dans la modification des nœuds et des arêtes des graphes. Lorsqu'un objet est saisi par le robot, il est supprimé du graphe de scène et transféré au graphe du robot. Inversement, lorsqu'un objet est déposé, il est réintégré au graphe de scène avec une mise à jour de sa position. Toutefois, cette mise à jour est centrée exclusivement sur les objets manipulables ou impliqués dans l'exécution de la tâche, sans prise en compte explicite de la traversabilité de l'environnement.

De manière similaire, SayPlan [61] exploite une représentation de l'environnement sous forme d'un graphe de scène 3D hiérarchique (3DSG) pour structurer la planification des tâches robotiques. Son approche repose sur un processus en deux étapes : une recherche sémantique et un replanning itératif. Dans un premier temps, SayPlan réduit la complexité de la scène en extrayant dynamiquement un sous-graphe pertinent pour la tâche en cours. SayPlan intègre également un mécanisme de correction basé sur une simulation du graphe de scène. Lorsque le plan généré contient des incohérences (ex. tentative de saisir un objet alors que le robot tient déjà quelque chose), une boucle de rétroaction itérative est déclenchée : le LLM reçoit un retour d'erreur textuel et reformule le plan en conséquence, jusqu'à obtenir une séquence d'actions réalisable.

Le travail de [62] propose une approche de mise à jour continue des cartes sémantiques pour des robots mobiles opérant dans des environnements domestiques dynamiques. La carte sémantique considérée est une représentation en 2D comprenant des éléments tels que les murs, les pièces et les obstacles temporaires. La mise à jour se fait de manière incrémentale lors des différentes missions du robot. Lors de chaque mission, une correspondance spatiale des éléments sémantiques précédemment connus est réalisée en

s'appuyant sur le système de localisation SLAM visuel. Une « Meta Map » temporaire est utilisée pour gérer et valider progressivement les changements détectés par le robot. Après validation, ces changements deviennent persistants dans la carte initiale, assurant une adaptation stable et durable aux environnements dynamiques. Toutefois, ces mises à jour restent limitées au sous-graphe spécifique à la tâche en cours et n'affectent pas la structure globale du 3DSG.

En revanche, les approches d'apprentissage de bout en bout [63, 56] évitent la conception explicite de cartes et reposent sur des représentations implicites pour la prédiction des actions. Bien qu'elles réduisent l'effort d'ingénierie, elles ne permettent pas de mises à jour explicites de la carte ni de mémoire persistante, ce qui limite leur efficacité dans les environnements nécessitant une adaptation à long terme.

Enfin, des approches de cartographie dynamique topologique ou basée sur des graphes, comme SOON [59] et RoboHop [53], vont au-delà de l'ObjectNav en construisant des cartes sémantiques évolutives sous forme de graphe de navigation. Ces méthodes permettent des requêtes avancées et un raisonnement relationnel, en ajoutant dynamiquement de nouveaux nœuds, en modifiant l'état des objets et en actualisant la connectivité des graphes de navigation.

3.2.5 Stratégies de replanification dans la navigation sémantique

Les approches traditionnelles [23, 41, 42] reposent sur des cartes prédéfinies et des algorithmes conventionnels comme A* ou l'algorithme de Dijkstra, en se concentrant principalement sur des ajustements locaux pour éviter les obstacles. En navigation robotique, on peut distinguer deux stratégies de replanification face aux imprévus environnementaux. Les stratégies de replanification ajustent la trajectoire d'un robot lorsqu'il rencontre des obstacles ou des changements dans l'environnement.

— Replanification de tâches (task-level replanning) : La première, la replanification de tâches, intervient au niveau décisionnel abstrait en réorganisant les objectifs ou actions symboliques sans modifier nécessairement la trajectoire physique. Des travaux comme SayPlan [61] corrigent itérativement les plans de haut niveau en exploitant les retours d'un simulateur de graphe de scène. Bien que ces approches améliorent l'organisation des séquences de tâches, elles ne réalisent pas de replanification au niveau du chemin ni d'ajustements globaux de trajectoire, car elles reposent sur des représentations statiques de l'environnement.

Ces méthodes excellent dans les environnements structurés où les contraintes sont principalement logiques, mais montrent leurs limites face à des obstacles physiques imprévus qui bloquent concrètement le passage du robot.

— Replanification globale du chemin : À l'inverse, la replanification globale du chemin se concentre spécifiquement sur le recalcul complet de la trajectoire physique lorsque des obstacles rendent le chemin initial impraticable, tout en conservant strictement le même

objectif final. Cette distinction opérationnelle est cruciale dans les environnements dynamiques réels où les blocages matériels (portes fermées, couloirs obstrués) sont fréquents.

Dans leur travail, [64] introduisent le concept de planification continue (lifelong planning) qui permet à un robot mobile de recalculer de façon répétée et optimale sa trajectoire au fur et à mesure que ses connaissances sur l'environnement changent. Contrairement à des planifications complètes à partir de zéro, leur méthode exploite les résultats précédents pour réduire considérablement le temps de calcul tout en garantissant l'optimalité des nouveaux chemins. Cette approche dite « lifelong » repose sur une variante incrémentale de l'algorithme A*, utilisée pour replanifier rapidement et de manière optimale la trajectoire du robot chaque fois qu'une modification de la carte est détectée.

[65] proposent une approche de navigation topologique visuelle reposant sur une représentation en graphe dont les nœuds sont des images RGB-D et les arêtes indiquent des déplacements locaux réalisables. Ce graphe est maintenu de manière dynamique en fonction des expériences de navigation accumulées par le robot, ce qui contribue à améliorer continuellement la robustesse et la qualité de la navigation.

3.2.6 Limites des approches existantes

Les approches existantes en navigation sémantique présentent certaines limites qui peuvent affecter leur efficacité dans des environnements dynamiques et complexes. Les méthodes basées sur l'exploration, bien qu'adaptées aux environnements inconnus, manquent souvent de représentations sémantiques robustes et peinent à exploiter efficacement la structure de l'environnement. À l'inverse, les approches utilisant des cartes statiques offrent une navigation efficace lorsque l'environnement est préalablement connu, mais deviennent rapidement obsolètes face à des changements imprévus, rendant difficile l'adaptation en temps réel.

Certaines méthodes introduisent des mises à jour dynamiques, mais ces mises à jour sont généralement temporaires et ne conservent pas de mémoire persistante, limitant leur capacité à gérer des évolutions progressives et récurrentes de l'environnement. Un autre défi majeur concerne la planification de trajectoires et le replanning face aux obstacles. Les approches traditionnelles reposent principalement sur des algorithmes comme A* ou Dijkstra, qui fonctionnent bien pour des environnements statiques, mais montrent leurs limites dans des contextes nécessitant des ajustements globaux de la trajectoire. De plus, la plupart des travaux existants manquent d'explicabilité dans les choix de navigation, ce qui limite leur adoption dans des applications industrielles ou collaboratives où la transparence des décisions est essentielle.

Enfin, les approches récentes basées sur les LLM, bien qu'innovantes, introduisent de nouvelles limitations. Ces modèles génèrent des plans sur la base de prédictions statistiques, ce qui peut engendrer des erreurs d'interprétation, notamment en présence d'instructions ambiguës ou de situations inhabituelles. Leur performance dépend fortement de la qualité du prompt fourni, rendant l'interface utilisateur sensible à la formulation des commandes. De plus, la plupart des systèmes LLM ne conservent pas de mémoire persistante : les tâches sont traitées comme indé-

pendantes, empêchant le robot d'apprendre des expériences passées ou de corriger ses erreurs de façon durable. Enfin, la confiance accordée aux décisions générées reste difficile à évaluer, ce qui soulève des enjeux en termes de fiabilité, de contrôle et d'acceptabilité dans des environnements critiques.

Le framework IntelliNav répond à ces défis pour permettre une navigation robuste et explicable. Ce système repose sur une Hierarchical Semantic Topometric Map (HSTM), une représentation hybride qui structure l'environnement en intégrant à la fois des informations topologiques, métriques et sémantiques. IntelliNav se distingue des approches existantes par sa capacité à mettre à jour la carte de manière dynamique, en ajustant les connexions entre espaces et en repositionnant les objets en fonction des changements observés. Il combine une navigation sur des trajectoires prédéfinies avec une exploration adaptative des objets non cartographiés, garantissant ainsi une navigation efficace dans des environnements partiellement connus. De plus, il intègre une mémoire persistante qui enregistre les expériences passées et influence les décisions futures, optimisant ainsi les performances du robot sur le long terme.

3.3 Framework IntelliNav : structure et fonctionnalités

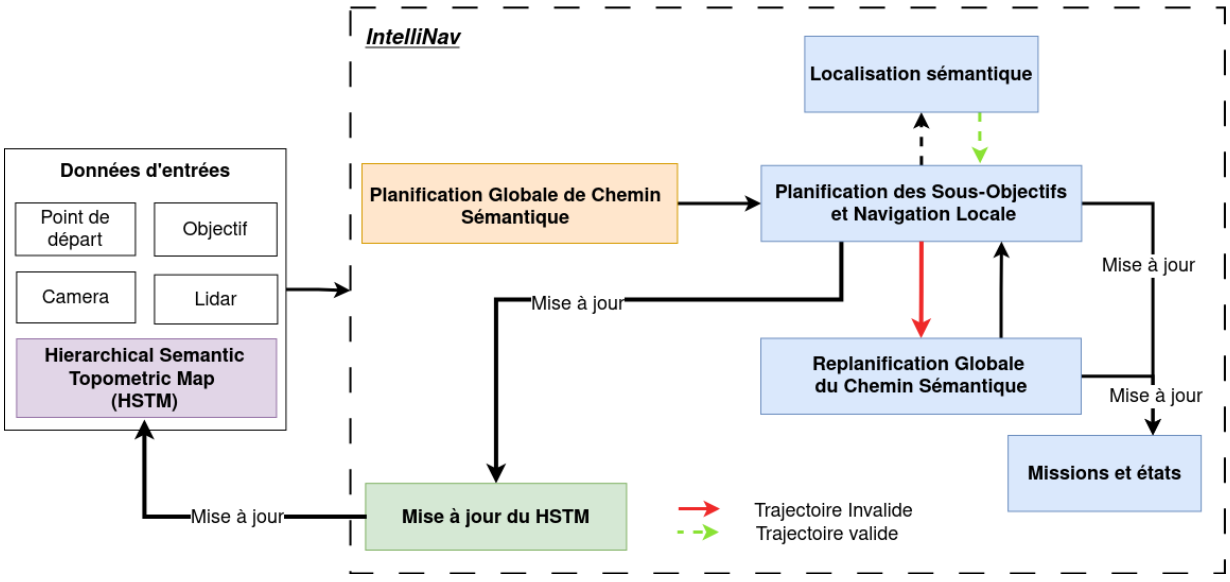


FIGURE 13 – IntelliNav Framework

Le framework IntelliNav présenté sur la figure 13 repose sur une architecture modulaire qui structure la navigation sémantique du robot en plusieurs étapes. Le système génère un chemin sémantique global reliant la position de départ à l'objectif final, tout en le divisant en sous-objectifs pour assurer une navigation progressive et explicable. Chaque sous-objectif est validé au fur et à mesure de la progression du robot, en s'appuyant sur la HSTM, qui est mise à jour en temps réel en fonction des observations effectuées. En cas d'obstacle empêchant l'atteinte d'un sous-objectif, le système déclenche un processus de replanification global dynamique, recalculant un chemin alternatif pour maintenir la continuité de la navigation. L'état de la mission du robot

est suivi en continu, assurant une gestion fluide des transitions entre les différentes étapes de la navigation. L'ensemble des modules d'IntelliNav collabore ainsi pour garantir une navigation efficace et explicable, où chaque action entreprise par le robot peut être justifiée en fonction de la structure de la carte et des événements rencontrés.

La suite de cette section décrit chaque module du framework IntelliNav, en commençant par la gestion des missions et des états du robot, puis en détaillant les étapes de la planification globale, de la navigation locale et de la mise à jour dynamique de la carte HSTM.

Ces modules sont illustrés dans les figures suivantes : la structure de la HSTM (figure 14), la planification sémantique globale (figure 15), le processus de mise à jour dynamique de la carte HSTM (figure 16) et la navigation locale basée sur la costmap (figure 17).

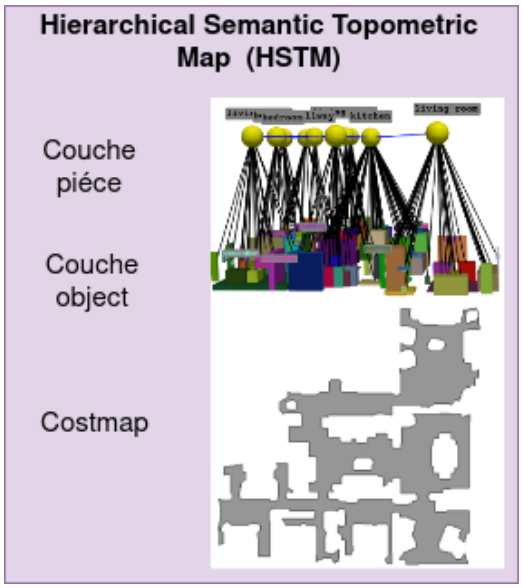


FIGURE 14 – Couche de la HSTM utilisé par IntelliNav

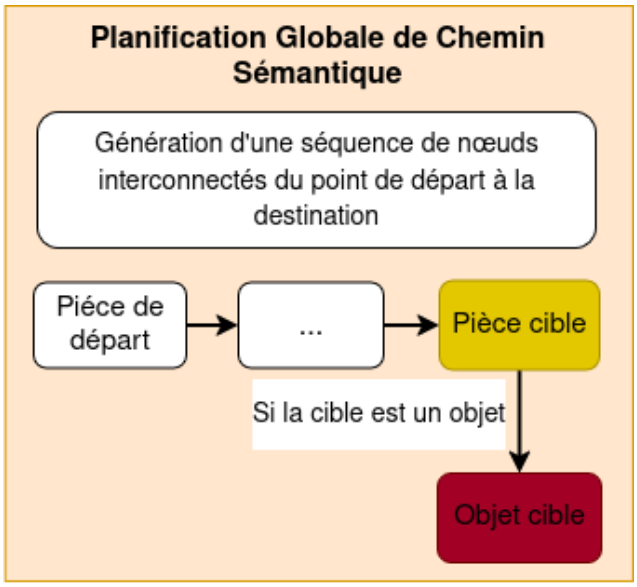


FIGURE 15 – IntelliNav : planification sémantique globale

3.3.1 Missions et états du robot

Dans le cadre d'IntelliNav, une mission désigne une tâche de navigation spécifique confiée au robot. Elle peut consister à naviguer vers un objet donné, à naviguer vers une pièce spécifique, ou encore à rechercher un objet non cartographié. Une mission est considérée comme terminée lorsque l'objectif est atteint et validé, soit par la détection de l'objet recherché, soit par l'arrivée dans la pièce cible. Si aucun chemin viable ne permet d'atteindre l'objectif ou si l'objet reste introuvable après exploration, la mission est déclarée échouée. Durant l'exécution de la mission, le robot traverse différents états, qui définissent sa progression depuis le début de la navigation jusqu'à la validation ou l'échec de la mission. Ces états structurent le comportement du robot et régissent les décisions prises à chaque étape du processus de navigation.

Ces mécanismes garantissent également une navigation explicable, en offrant une vue claire des décisions prises par le robot à chaque étape de la mission. La gestion des états et des



FIGURE 16 – IntelliNav : description des mises à jour de la HSTM

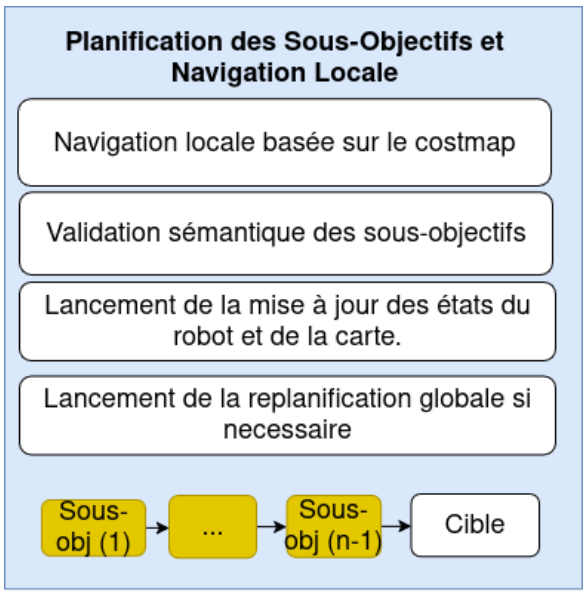


FIGURE 17 – IntelliNav : Navigation locale

transitions entre ces états est au cœur du système de navigation. Elle rend le robot capable de s'adapter aux changements imprévus tout en maintenant une transparence opérationnelle.

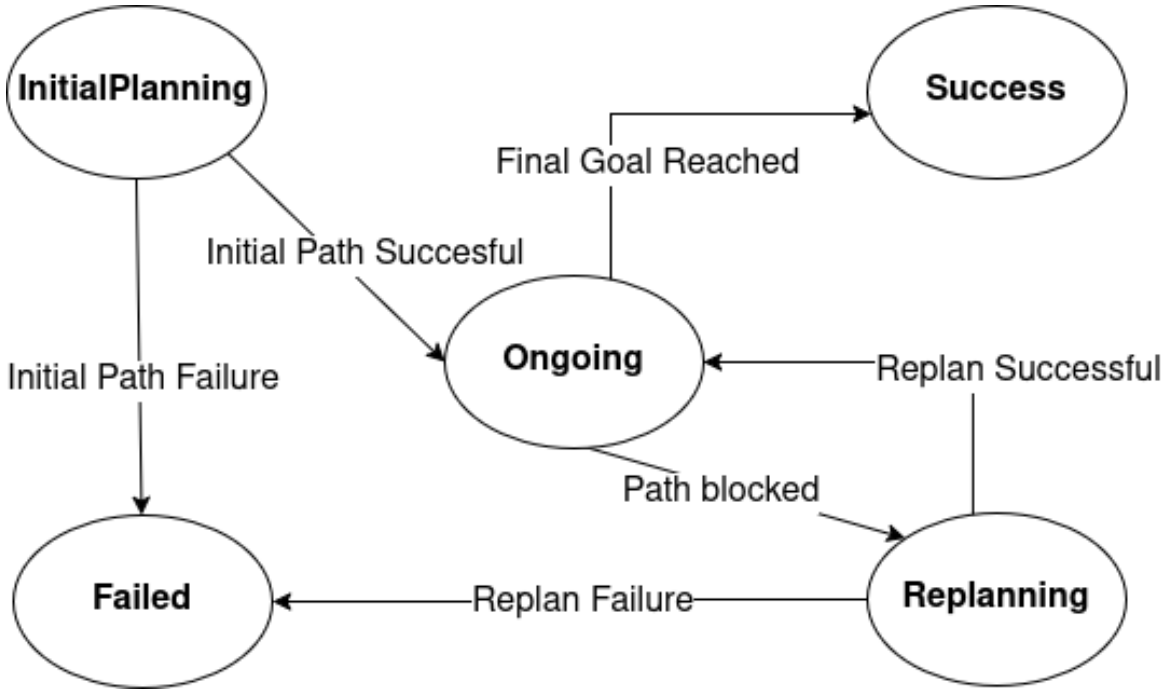


FIGURE 18 – Les differents états du robot lors d'une mission

3.3.1.1 Les differents états définis dans IntelliNav

Le processus de navigation dans IntelliNav est structuré autour de quatre états principaux, qui reflètent l'évolution de la mission et les ajustements nécessaires en fonction des observations du robot.

— InitialPlanning : Cet état correspond à la phase de calcul initial du chemin global sémantique, effectué à partir de la position actuelle du robot jusqu'à l'objectif final N_g. Cette étape prépare la séquence de sous-objectifs (subgoals) à suivre avant d'entrer en exécution.
— Ongoing : Cet état indique que le robot est en cours d'exécution de la navigation. Il suit le chemin sémantique global défini et valide chaque sous-objectif (subgoal) atteint. Pendant cet état, le robot effectue une planification locale basée sur la costmap tout en surveillant les changements environnementaux. Par exemple, si le robot détecte un obstacle mineur, il ajuste sa trajectoire localement sans nécessiter une replanification globale.
— Replanning : Cet état est activé lorsqu'un obstacle permanent empêche le robot d'atteindre un sous-objectif, comme un chemin bloqué. IntelliNav met à jour la structure HSTM pour refléter l'état actuel de l'environnement, et un nouveau chemin sémantique global est calculé en tenant compte des nouvelles contraintes. Par exemple, si une porte est fermée, le robot recalcule un chemin alternatif pour atteindre son objectif.
— Success : L'état final lorsque le robot atteint son objectif avec succès. La mission est alors considérée comme accomplie.
— Failed : L'état final lorsque le robot ne peut pas poursuivre la mission, soit en raison de l'absence d'un chemin viable, soit à cause d'obstacles insurmontables identifiés pendant la navigation. Par exemple, si tous les chemins vers l'objectif sont bloqués et qu'aucune alternative n'est trouvée, la mission est marquée comme échouée.

3.3.1.2 Transitions entre états

Les transitions entre ces états sont déclenchées par des événements observés lors de la navigation. Ces transitions permettent au robot de réagir de manière dynamique aux changements environnementaux tout en maintenant une cohérence dans l'exécution de la mission. Les transitions dans IntelliNav sont :

— Initial Path Successful : Le chemin global sémantique initial a été généré avec succès à partir de la position de départ. La mission peut commencer et le robot passe à l'état *Ongoing*.
— Initial path failure : L'impossibilité de générer un chemin initial à partir de la position actuelle entraîne un échec immédiat de la mission. Par exemple, si aucun chemin viable n'est trouvé entre la position de départ et la cible (carte incomplète ou contraintes bloquantes), l'état passe immédiatement à *Failed*.
— Final Goal Reached : Lorsqu'un sous-objectif ou l'objectif final est atteint, l'état passe à *Success*. Par exemple, si le robot atteint la pièce cible ou localise l'objet recherché, la mission est considérée comme réussie.
— Path Blocked : Lors de la navigation vers un sous-objectif, une obstruction persistante est détectée sur le chemin. Le robot déclenche le *Replanning* pour tenter de recalculer

un nouveau chemin valide.

— Replan successful : Si un nouveau chemin viable est trouvé après recalcul, le robot revient à l'état Ongoing et poursuit sa mission.

— Replan Failure : Si aucun chemin viable n'est possible après recalcul, la mission est marquée comme échouée, et l'état passe à *Failed*. Par exemple, si le robot atteint la pièce cible ou localise l'objet recherché, la mission est considérée comme réussie.

Les transitions entre ces états sont déterminées par des événements de navigation, tels que la détection d'obstacles, l'échec de la replanification, ou l'atteinte de l'objectif. Ce système distingue si le robot atteint son objectif via le chemin initial ou un chemin replanifié, offrant ainsi une explication claire des changements de chemin et de l'état de la mission. La définition des états spécifiques dans IntelliNav joue un rôle essentiel dans l'optimisation de la navigation sémantique adaptative et explicable. De plus, cette structuration renforce l'explicabilité du processus de navigation : chaque transition entre les états est accompagnée d'une justification claire des décisions prises, augmentant ainsi la transparence et la confiance dans le système. Ces états et leurs transitions constituent un pilier fondamental pour gérer les environnements dynamiques, illustrant une approche pensée pour la fiabilité opérationnelle et l'explicabilité dans des contextes complexes et réels.

3.3.2 Localisation sémantique

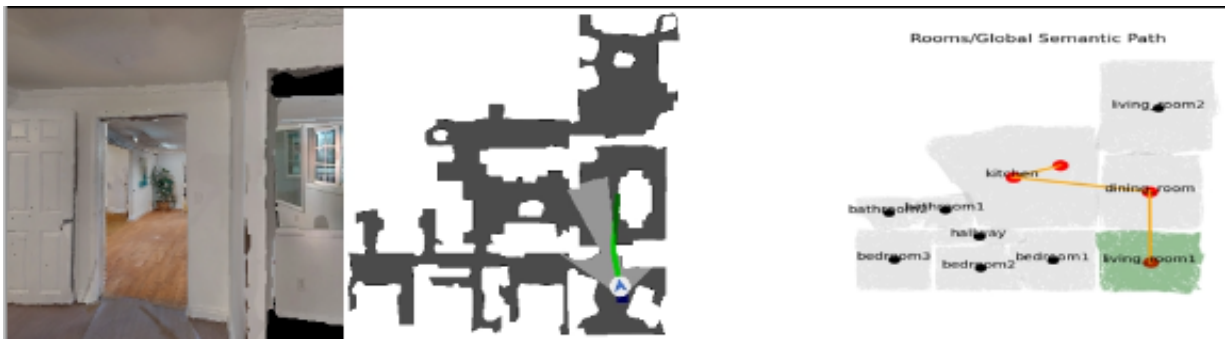
3.3.2.1 Introduction

La localisation sémantique permet d'assurer que le robot comprend non seulement où il se trouve lors de la navigation, mais aussi le contexte fonctionnel de cet emplacement. Dans le framework IntelliNav, la localisation sémantique, étroitement liée à la HSTM, identifie la pièce dans laquelle se trouve le robot, en allant au-delà d'une simple position géométrique. Contrairement à une localisation purement métrique, qui repose uniquement sur des coordonnées spatiales, la localisation sémantique associe la position du robot à une pièce spécifique en tenant compte des informations sur les objets, des limites des pièces et de leurs relations avec les pièces environnantes. Cela permet au robot de raisonner sur des concepts abstraits comme « je suis dans la salle de réunion » ou « cuisine », facilitant ainsi des prises de décision contextuellement adaptées.

Les images sur la figure 19 illustrent deux robots positionnés à des endroits distincts d'un environnement intérieur, et leur correspondance avec les pièces de la HSTM.

La capacité à confirmer que le robot se trouve dans la pièce correcte est essentielle pour garantir la validation des sous-objectifs au cours de la navigation. En effet, les sous-objectifs de la mission correspondent à des pièces ou des emplacements spécifiques. La validation précise de la position du robot dans la bonne pièce confirme l'atteinte d'un sous-objectif et déclenche la transition vers le suivant.

Cette capacité est particulièrement importante dans les scénarios d'exploration d'objets non cartographiés, où l'identification correcte des pièces permet d'optimiser la recherche. Lorsqu'un



(a) Localisation sémantique - Salon (living_room1)



(b) Localisation sémantique - Salle à manger (dining_room)

FIGURE 19 – Exemple de localisation sémantique : Correspondance entre la position du robot et la pièce associée.

objet cible peut être situé dans plusieurs pièces potentielles, le robot doit être capable de détecter avec précision son entrée dans chacune d'elles afin de déterminer si elles correspondent aux zones les plus pertinentes pour la recherche. Cela permet d'activer l'exploration ciblée uniquement lorsque le robot atteint une pièce susceptible de contenir l'objet, évitant ainsi l'exploration complète d'espaces non pertinents.

3.3.2.2 Processus de localisation avec IntelliNav

La localisation sémantique réduit les erreurs en combinant des approches géométriques et sémantiques, rendant la navigation plus fiable. Le processus de localisation sémantique dans IntelliNav repose sur deux étapes principales : la validation géométrique et la validation sémantique. Ces étapes sont complémentaires et permettent au robot de confirmer sa position.

— *Validation de la localisation géométrique des pièces* : La première étape de la localisation sémantique consiste à valider la position du robot par rapport aux limites géométriques des pièces définies dans la couche des pièces de la HSTM. Chaque pièce est représentée par un polygone P en 2D, défini par un ensemble de sommets (x1,y1),(x2,y2),...,(xn,yn) stockés dans la carte HSTM. Le robot, dont la position est représentée par un point (xr,yr), doit déterminer s'il se trouve à l'intérieur de l'un de ces polygones. L'appartenance du point (xr,yr) au polygone représentant une pièce est vérifiée à l'aide d'un test d'inclusion de point dans un polygone. Si l'appartenance est confirmée, la validation

géométrique est réussie et la localisation du robot est validée.

Cette approche garantit une localisation précise dans les environnements où les limites physiques des pièces sont bien définies. En effet, plus les frontières des pièces sont clairement définies dans la carte, plus la validation de la présence du robot dans une pièce donnée est rapide et fiable.

— *Validation de la localisation sémantique par détection d'objets* : Pour renforcer la validation géométrique, le robot effectue une analyse des objets présents dans son champ de vision. Cette seconde validation permet de confirmer ou d'infirmer l'estimation géométrique en comparant les objets perçus avec ceux attendus dans la pièce présumée, tels qu'ils ont été enregistrés lors de la phase de cartographie. Cette approche repose sur l'utilisation de modèles de détection d'objets, qui peuvent être similaires à ceux employés lors du processus de cartographie initiale. Des modèles de vision par ordinateur tels que YOLO (You Only Look Once)[66], Faster R-CNN [67, 68] ou [69, 70] peuvent être utilisés pour identifier et classifier les objets détectés lors de la navigation. Ces modèles doivent s'appuyer sur les mêmes classes d'objets que celle définies dans la HSTM afin de garantir une correspondance cohérente entre les objets perçus et ceux enregistrés dans la carte sémantique. La figure 20 illustre la détection d'objet.



FIGURE 20 – Détection d'objet

Le robot compare les objets détectés autour de lui à ceux stockés dans la couche des objets de la HSTM. Une correspondance entre plusieurs objets observés et leur emplacement attendu dans la carte peut renforcer la confiance dans la localisation. Pour assurer une validation de la présence d'un robot dans une pièce, le système exige généralement la détection d'au moins trois objets distincts dont les types et les positions correspondent aux données de la HSTM. Cette méthode permet d'écarter les ambiguïtés pouvant résulter de détections erronées ou de changements mineurs dans l'environnement.

Ce mécanisme est particulièrement utile dans des environnements où les frontières géométriques peuvent être ambiguës.

La localisation sémantique dans IntelliNav illustre la manière dont des représentations riches et multi-couches, comme la HSTM, peuvent transformer la navigation robotique en offrant une compréhension contextuelle et une adaptabilité accrues. Ce processus constitue une base solide pour les étapes ultérieures de planification globale et locale, renforçant ainsi l'efficacité du framework IntelliNav.

3.3.3 Planification globale de chemin sémantique

La planification globale sémantique dans IntelliNav repose sur la structure multi-couches de la HSTM.

Le chemin global sémantique sur IntelliNav est défini comme une séquence ordonnée de noeuds (pièce ou objet) interconnectés. Ainsi, la planification globale génère un itinéraire structuré qui guide le robot depuis sa position initiale jusqu'à son objectif, en exploitant les connexions entre les pièces stockées dans la HSTM. Cette organisation hiérarchique permet une navigation robuste, où chaque étape du chemin est définie en fonction de la structure de l'environnement et des connaissances sémantiques préexistantes. La figure 21 illustre un chemin global sémantique d'un environnement.

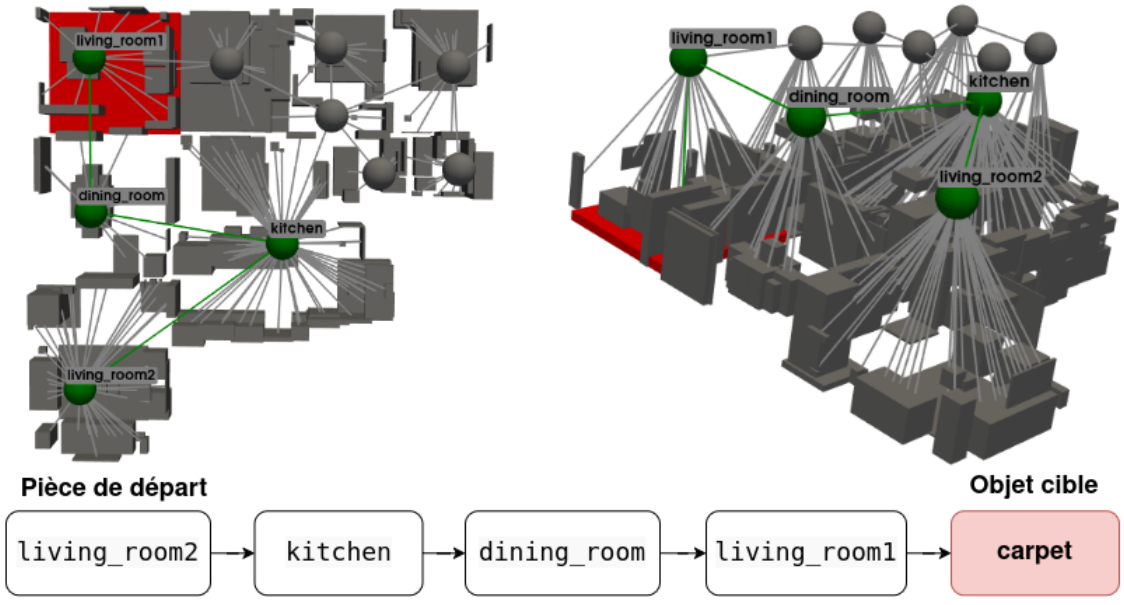


FIGURE 21 – Planification globale du chemin

L'objectif principal est de générer un chemin sémantique global reliant un point de départ à un objectif , en tenant compte des contraintes spatiales et contextuelles de la navigation.

L'algorithme commence par l'identification du type d'objectif, qui peut être de deux natures : pièce ou objet. Lorsque la mission consiste à atteindre une pièce spécifique, le chemin global est une succession de pièces à traverser jusqu'à la destination. En revanche, si l'objectif est un objet particulier, la planification globale inclut à la fois la liste des pièces à traverser puis l'objet

cible à localiser une fois la pièce atteinte. Le chemin sémantique est généré sous forme d'une séquence de nœuds à partir des couches des pièces et des objets de la HSTM. Cette planification utilise l'algorithme de Dijkstra appliqué sur un graphe pondéré, garantissant une optimisation du chemin en minimisant les coûts cumulés, tout en intégrant les contextes sémantiques et l'état dynamique de l'environnement. Les contextes sémantiques assurent des routes pertinentes en fonction de la classification des pièces et des objets. L'état dynamique de l'environnement tient compte des poids des connexions entre les pièces définis sur la HSTM.

Une fois le chemin global généré, la navigation est exécutée en progressant séquentiellement à travers les sous-objectifs définis dans la HSTM. L'algorithme suit une approche modulaire avec trois modes de navigation présentés sur l'algorithme 2 : Mode Exploration Sémantique, Mode Navigation Ciblée, Mode Exploration Multi-cibles. Les sections suivantes détaillent les scénarios spécifiques de planification sémantique globale.

Algorithm 2 Planification Globale Sémantique avec IntelliNav

Entrées : $HSTM$, N_s, N_g ▷ Carte sémantique, nœud de départ, nœud cible
Sortie : *chemin_optimal*
procedure SemanticGlobalPlanning($HSTM$, N_s, N_g)
    chemin_optimal $\leftarrow \emptyset$ ▷ Initialisation du chemin
    état_objectif $\leftarrow$ TrouverÉtatObjectif($HSTM, N_g$)
    if *état_objectif* est vide **then** ▷ Objectif non cartographié
        Mode Exploration Sémantique :
        objectif_prévu $\leftarrow$ PrédictionLLM($HSTM, N_g$) ▷ LLM prédit les emplacements les plus probables
        chemin_optimal $\leftarrow$ Dijkstra($HSTM, N_s$, *objectif_prévu*)
        return *chemin_optimal*
    else if longueur(*état_objectif*) $= 1$ **then** ▷ Objectif unique et connu
        Mode Navigation Ciblée :
        chemin_optimal $\leftarrow$ Dijkstra($HSTM, N_s, N_g$) ▷ Calcul du plus court chemin
        return *chemin_optimal*
    else ▷ L'objectif peut exister à plusieurs endroits
        Mode Exploration Multi-cibles :
        Initialiser *longueur_min_chemin* $\leftarrow \infty$
        for chaque N_{g_i} dans *état_objectif* **do** ▷ Parcourir toutes les instances connues de l'objectif
            chemin $\leftarrow$ Dijkstra($HSTM, N_s, N_{g_i}$)
            if *chemin* est valide et *chemin.length*() $<$ *longueur_min_chemin* **then**
                chemin_optimal $\leftarrow$ *chemin*
                longueur_min_chemin $\leftarrow$ *chemin.length*()
            end if
        end for
        return *chemin_optimal*
    end if
end procedure

3.3.3.1 Scénario 1 : navigation vers un objet ou lieu déjà cartographié

Lorsqu'une mission est assignée au robot dans ce scénario, la première étape consiste à identifier précisément l'objectif à atteindre. Ce dernier peut être une pièce spécifique, comme une cuisine ou un bureau, ou bien un objet précis, tel qu'un ordinateur ou une tasse. Dans le premier cas, la navigation se concentre sur les relations topologiques entre les pièces, tandis que dans le second cas, une recherche d'objet localisée au sein d'une pièce donnée est nécessaire. Une fois la cible identifiée, le système extrait la position du nœud correspondant et génère une requête de planification qui définit la position actuelle du robot comme nœud de départ et la cible comme nœud d'arrivée, tout en considérant les relations de connectivité existantes.

L'algorithme de planification exploite l'algorithme de Dijkstra pour calculer le chemin optimal dans le graphe pondéré de la HSTM. Le coût de chaque arête reflète sa traversabilité, tandis que les nœuds sont pondérés en fonction de la présence des objets associés et de l'historique de visite du robot. En minimisant le coût cumulé des arêtes traversées, cette méthode garantit une navigation fluide et efficace. Lorsqu'une alternative existe, l'algorithme priorise les chemins marqués comme non obstrués ou présentant un risque réduit, en exploitant la mémoire intégrée dans la HSTM. Cette capacité permet au robot de s'adapter dynamiquement aux modifications de l'environnement.

3.3.3.2 Scénario 2 : exploration pour des objets non cartographiés

Dans ce scénario, l'objectif consiste à localiser un objet cible qui n'est pas préalablement enregistré dans la HSTM. Par exemple, si le robot recherche une tasse, les pièces étiquetées comme « cuisine » ou « salle à manger » seront priorisées. Pour identifier les pièces les plus probables où se trouve un objet recherché, le système extrait de la HSTM la liste des nœuds « pièce » avec leur attributs sémantiques , puis interroge un modèle de type ChatGPT à l'aide d'un prompt structuré. Celui-ci évalue, pour chaque pièce, la probabilité d'y trouver l'objet ciblé. Seules les pièces dont le score de pertinence dépasse un seuil (par exemple 80%) sont conservées, afin de limiter les hypothèses et d'améliorer la fiabilité de la prédiction. Le LLM peut, par exemple, suggérer que la cuisine est le lieu le plus probable pour trouver une tasse.

Une fois les emplacements probables identifiés, l'algorithme de Dijkstra est employé pour calculer les chemins optimaux vers ces pièces, en tenant compte des poids dynamiques des connexions dans la HSTM. Le robot explore ensuite ces lieux dans un ordre de probabilité décroissante. Chaque validation d'hypothèse entraîne une mise à jour en temps réel de la HSTM, indiquant si l'objet recherché a été trouvé ou si le lieu a été complètement exploré.

Cette méthode permet au robot d'explorer des environnements partiellement inconnus. Elle combine l'exploitation des données contextuelles fournies par la HSTM avec une exploration active guidée par des prédictions sémantiques, augmentant ainsi les chances de succès dans des missions complexes.

Bien que ChatGPT ait été utilisé dans nos expérimentations, d'autres modèles de LLM peuvent également être intégrés, tels que Gemini [71], Mistral [72], ou encore DeepSeek [73],

en fonction des contraintes d'infrastructure, des préférences d'architecture ou des performances attendues sur des tâches spécifiques.

3.3.3.3 Scénario 3 : exploration multi-cibles

Dans les scénarios où l'objectif recherché peut être présent à plusieurs endroits différents dans l'environnement, IntelliNav adopte une stratégie d'exploration multi-cibles basée sur la hiérarchie de la HSTM. Ce mode permet au robot d'optimiser son trajet en explorant de manière intelligente les différentes occurrences potentielles de la cible, tout en minimisant les déplacements inutiles.

Lorsque plusieurs positions candidates pour l'objectif sont identifiées dans la HSTM, le robot sélectionne en priorité l'instance la plus proche en fonction du coût de déplacement minimal. Cette sélection repose sur le graphe pondéré de la HSTM, où les poids des connexions entre les pièces et les poids des nœuds reflètent les contraintes de navigation.

Lorsque le robot atteint une des positions candidates, il procède à une validation locale pour confirmer la présence de l'objet recherché. Si l'objectif est trouvé, la mission est déclarée réussie et la HSTM est mise à jour pour enregistrer la localisation validée de l'objet. Dans le cas contraire, l'emplacement est marqué comme exploré, afin d'éviter d'y revenir inutilement.

Ce processus se répète itérativement jusqu'à ce que l'objet soit localisé ou que toutes les positions candidates aient été explorées. Si après avoir testé toutes les occurrences possibles, le robot ne trouve pas l'objectif, la mission est déclarée échouée. le mode exploration multi-cibles assure une navigation efficace en évitant les déplacements superflus et en maximisant les chances de succès dans la recherche d'objets.

La planification globale sémantique avec IntelliNav démontre son efficacité dans des environnements complexes, qu'ils soient entièrement cartographiés ou partiellement connus. La combinaison de la HSTM, de l'algorithme de Dijkstra et des capacités prédictives des LLM permet une navigation robuste, explicable et adaptée à des scénarios réels. Cette méthode pose les bases pour une navigation locale et un replanning efficace, qui seront explorés dans les sections suivantes.

3.3.4 Planification des sous-objectifs et navigation locale

La navigation locale dans IntelliNav repose sur une progression itérative structurée, où le robot franchit chaque étape du chemin sémantique global tout en s'adaptant dynamiquement aux conditions environnementales. Cette approche garantit une transition fluide vers l'objectif final, avec une gestion efficace des sous-objectifs définis par le chemin global. Dès que le chemin global $P = [N_1, N_2, \ldots, N_g]$ est généré, le robot débute son déplacement à partir du premier nœud N_1, correspondant à sa position initiale. Il progresse ensuite séquentiellement vers les sous-objectifs N_{i+1}, $0 \leq i \leq g-1$, en validant chaque étape avant d'atteindre la destination finale N_g.

3.3.4.1 Navigation locale basée sur une costmap

Cette approche conduit à une navigation efficace, en combinant connaissances sémantiques, géométriques et mise à jour continue des cartes HSTM. Elle favorise une prise de décision explicable tout au long de la mission. La décomposition du chemin global en sous-objectifs permet de suivre précisément la progression du robot. Chaque transition entre deux nœuds de la HSTM est identifiable, ce qui rend possible la localisation explicite des étapes réussies ou des points de blocage. La progression structurée facilite non seulement l'adaptation dynamique en cas d'échec local, mais aussi l'interprétation du comportement global du robot par un opérateur humain.

La navigation locale d'IntelliNav repose sur l'utilisation d'une costmap qui encode les contraintes métriques de l'environnement. Cette costmap joue un rôle central dans la planification locale. Lorsqu'un sous-objectif est défini, le robot doit choisir la trajectoire optimale pour s'y rendre. Pour cela, l'algorithme de Dijkstra ou A* est appliqué sur la costmap, prenant en compte non seulement la distance minimale, mais aussi les zones de coût élevé représentant des obstacles potentiels. le robot peut ainsi ajuster sa trajectoire localement sans déclencher de replanification globale. Grâce à ce mécanisme illustré sur la figure 22, le robot est capable de réajuster son itinéraire en temps réel, garantissant une progression fluide.

En limitant les détours inutiles et en privilégiant les trajets les plus courts selon l'état actuel de l'environnement, cette approche assure une navigation adaptée aux contraintes spatio-temporelles.

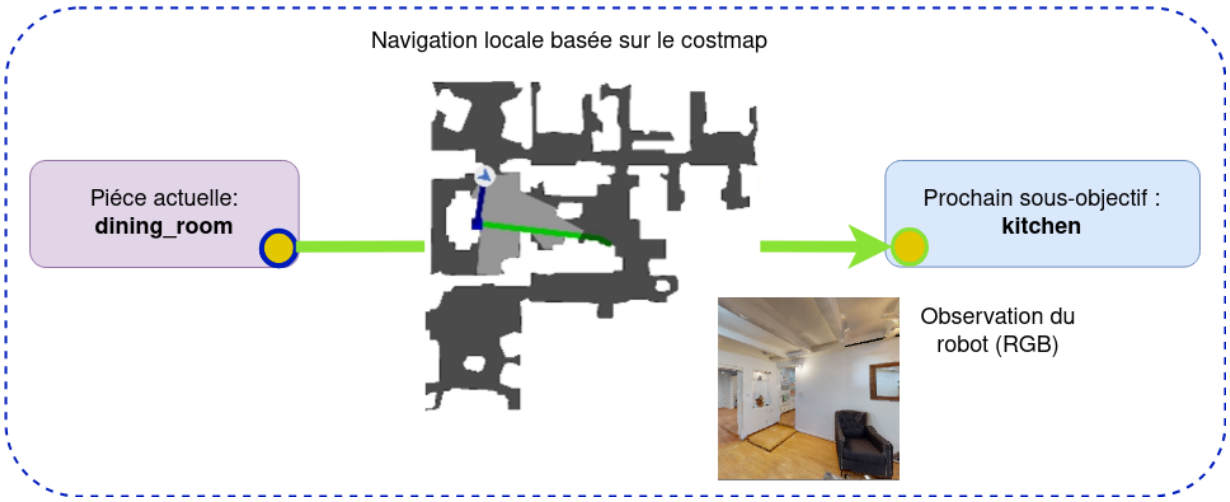


FIGURE 22 – Exemple de navigation locale entre deux pièces sur une costmap

Pendant la navigation, les capteurs du robot détectent les changements environnementaux et ajustent dynamiquement la HSTM. Par exemple, si une porte censée être ouverte est fermée, l'algorithme modifie l'état du chemin dans la HSTM et active un recalcul du chemin global. Grâce à ce mécanisme de mise à jour, le robot évite les trajets bloqués et optimise continuellement son itinéraire.

3.3.4.2 Validation d'un sous-objectif

IntelliNav intègre un mécanisme de validation des pièces basé sur la localisation sémantique qui permet d'optimiser la progression du robot et de réduire les distances inutiles. Dans un scénario de navigation sémantique classique, les robots utilisent généralement des graphes de navigation où chaque pièce est associée à un ou plusieurs nœuds spécifiques servant de points de validation. Pour confirmer qu'ils ont bien atteint une pièce donnée, ils doivent souvent naviguer jusqu'à un nœud central ou un ensemble de nœuds définis dans la représentation spatiale de l'environnement. Bien que structurée, cette approche entraîne des déplacements superflus, car le robot ne peut pas enchaîner directement les étapes de sa mission sans atteindre ces points prédéfinis.

À l'inverse, dans la navigation humaine, lorsqu'une personne entre dans une pièce, elle reconnaît immédiatement son environnement et ajuste instinctivement sa trajectoire en fonction de la disposition des lieux. Par exemple, si l'objectif est de traverser plusieurs pièces pour atteindre un point précis, elle ne se dirige pas systématiquement vers le centre de chaque pièce avant d'accéder à la porte suivante. Dès son entrée, elle perçoit son environnement, identifie la nature de la pièce et se dirige directement vers la sortie appropriée, optimisant ainsi son déplacement.

De même, une fois la validation de pièce effectuée, le robot peut directement planifier son passage au sous-objectif suivant, sans nécessiter un déplacement superflu vers un point central. Cette approche réduit considérablement les distances parcourues et améliore l'efficacité de la navigation en environnement dynamique.

3.3.4.3 Cas particulier : navigation vers un objet

Lorsque l'objectif final est un objet N_g, le robot commence par identifier la pièce où l'objet est censé se trouver (noœud N_{g-1}). Il suit alors le chemin sémantique global pour y parvenir, en s'appuyant sur la HSTM pour optimiser son trajet. Une fois dans la pièce cible, le robot entre dans une phase de recherche locale. Il utilise ses capteurs de vision et les informations spatiales de la HSTM pour repérer l'objet. Cette étape repose sur un modèle de reconnaissance d'objets, grâce auquel le robot analyse les caractéristiques visuelles de son environnement et peut confirmer la présence de l'objectif recherché. Dès que l'objet est détecté, la mission est déclarée réussie. Si l'objet n'est pas trouvé à l'emplacement prévu, le système soit relance une recherche dans la pièce, soit active le mode exploration multi-cible si l'objet est cartographié plusieurs fois pour explorer d'autres emplacements possibles, soit la mission est déclarée en échec.

3.3.4.4 Navigation locale lors des transitions inter-étages

Lorsque le prochain sous-objectif Ni+1 se situe à un étage différent du nœud actuel Ni, la navigation locale ne peut pas s'appuyer sur la costmap. En effet, les données de coût sont généralement valables pour un seul étage, et la structure verticale du bâtiment (escaliers, ascenseurs,

rampes) n'est pas représentée dans la costmap bidimensionnelle.

Il est essentiel de souligner que la navigation inter-étages dépend fortement de deux facteurs : la configuration de l'environnement, notamment la présence d'escaliers ou d'ascenseurs, et les capacités physiques et perceptives du robot. Par exemple, un robot capable de reconnaître visuellement les boutons d'un ascenseur et d'interagir physiquement avec eux requiert une stratégie de contrôle différente de celle d'un robot conçu pour monter des escaliers.

Dans le cas spécifique des transitions basées sur des escaliers, IntelliNav exploite des séquences de points de passage inter-étages pré-calculées lors de la phase de cartographie. Ces séquences sont ensuite réutilisées pendant la navigation pour reproduire la même trajectoire inter-étages. Cette approche repose sur l'hypothèse que la géométrie de la zone d'escalier reste inchangée entre les phases de cartographie et d'exécution.

Pour suivre ces trajectoires, des contrôleurs bas-niveau tels que Pure Pursuit [74] ou MPC (Model Predictive Control) [75] peuvent être employés afin de générer les commandes de mouvement à partir de la séquence de points. La détection d'obstacles sur les escaliers est gérée de manière implicite via le retour d'exécution : si le contrôleur échoue à atteindre un point de passage — soit parce qu'aucune commande valide ne peut être calculée, soit en raison d'une collision physique — la trajectoire est considérée comme temporairement inaccessible. L'arête correspondante dans la carte HSTM est alors marquée comme *bloquée*.

La navigation locale dans IntelliNav apporte plusieurs avantages significatifs. Tout d'abord, elle offre une capacité d'adaptation dynamique face aux changements imprévus de l'environnement, assurant ainsi une progression fluide et ininterrompue. De plus, son efficacité repose sur l'intégration de validations sémantiques et géométriques, qui réduisent considérablement les déplacements inutiles et optimisent les trajets. Enfin, l'explicabilité du processus se traduit par des décisions transparentes, directement liées aux informations extraites de la HSTM, offrant une compréhension claire des actions entreprises. En ce sens, la navigation locale complète efficacement la planification globale. Elle rend le robot apte à évoluer dans des environnements dynamiques.

3.3.5 Mise à jour de la carte sémantique HSTM

Dans des environnements intérieurs dynamiques, les modifications fréquentes, telles que les déplacements d'objets, les changements de connectivité entre les pièces, ou encore les obstacles temporaires, rendent essentiel l'ajustement en temps réel des cartes utilisées pour la navigation. Le framework IntelliNav intègre un processus de mise à jour dynamique de la HSTM garantissant ainsi une navigation fiable, adaptative et contextuellement informée. Grâce à ces mises à jour, le robot s'adapte aux conditions environnementales et améliore ses performances futures en exploitant les informations historiques. À chaque navigation locale, qu'elle soit validée ou non, la HSTM est mise à jour pour intégrer les nouvelles observations réalisées. Cette approche conduit à un enrichissement progressif de la représentation de l'environnement par le robot, favorisant une navigation à la fois explicable et adaptative. Sans ces mises à jour, le robot

risquerait de s'appuyer sur des informations obsolètes, entraînant des échecs de navigation ou des décisions inefficaces.

Les mises à jour de la HSTM concernent principalement deux aspects fondamentaux :

— Les relations entre les pièces, représentées par les arêtes ($E_{room\text{-}room}$), qui définissent les connexions navigables entre les différentes zones de l'environnement.

— Les attributs des objets et des pièces, représentés par les noeuds (N_{room}, N_{obj}), qui stockent les informations sémantiques et spatiales nécessaires pour la navigation.

Ces mises à jour sont effectuées de manière automatique et incrémentale en fonction des observations du robot lors de ses déplacements lors de la mission.

3.3.5.1 Mise à jour des états des arêtes

Les arêtes $E_{room\text{-}room}$ représentent les connexions navigables entre les différentes pièces. Leur mise à jour influence directement la planification des trajectoires, en ajustant le poids des chemins en fonction de leur accessibilité et de leur fiabilité.

— État 0 : Non testé (Untested). Il s'agit d'un chemin qui n'a pas encore été exploré ou validé. Lorsqu'un robot découvre un nouveau passage entre deux pièces, l'arête correspondante est initialement marquée comme non testée.

— État 1 : Validé (Validated). Une connexion est considérée comme validée lorsque le robot l'a traversée avec succès. Ce marquage indique que le chemin est fiable et qu'il peut être utilisé en toute sécurité pour les navigations futures.

— État 2 : Potentiellement bloqué (Potentially Blocked). À la fin d'une mission, les arêtes bloquées sont requalifiés comme "potentiellement bloquées", offrant un contexte historique. Ce statut est particulièrement utile pour gérer des chemins dont l'accessibilité peut varier (exemple : une porte qui est parfois ouverte et parfois fermée). Un chemin potentiellement bloqué peut être utilisé si aucune autre alternative viable n'est disponible.

— État 3 : Bloqué (Blocked). Lorsqu'un robot détecte un obstacle permanent, tel qu'une porte fermée ou un couloir obstrué, l'arête correspondante est marquée comme bloquée. Les chemins bloqués sont temporairement exclus des calculs de planification au cours de la mission jusqu'à ce qu'ils soient réévalués.

Chaque état est associé à un poids spécifique influençant la sélection des chemins lors de la planification globale. Une arête de statut "validé" possède un poids minimal, la rendant prioritaire lors du calcul des chemins. Une arête de statut "potentiellement bloqué" reçoit un poids intermédiaire, ce qui limite son utilisation aux cas où aucune autre solution n'est envisageable. Une arête de statut "bloqué" possède un poids maximal, excluant son utilisation par l'algorithme de navigation, sauf en cas de mise à jour ultérieure. Dans le cas des transitions inter-étages, des poids supplémentaires sont appliqués afin de refléter la complexité accrue des déplacements verticaux.

Dans un exemple de navigation avec IntelliNav, nous avons défini des poids spécifiques pour

chaque état des connexions entre pièces, comme présenté dans le tableau 3.1. Ces poids ont été soigneusement ajustés pour refléter les différentes conditions de navigabilité et influencer la planification des trajectoires de manière optimale.

Le poids attribué à une connexion bloquée est fixé à 999 afin de garantir son exclusion complète du calcul de trajectoire. En raison de ce coût extrêmement élevé, l'algorithme de planification basé sur Dijkstra évite systématiquement ces chemins à moins qu'aucune autre alternative ne soit disponible.

À l'inverse, le poids attribué aux connexions potentiellement bloquées est fixé à 10, afin de ne pas exclure totalement ces trajets de la planification. Lorsque plusieurs transitions sont nécessaires entre les pièces, l'algorithme peut comparer les coûts cumulés et sélectionner un itinéraire intégrant une connexion potentiellement bloquée si celle-ci réduit le coût total du parcours. Ainsi, IntelliNav maintient une certaine flexibilité, en laissant la possibilité au robot d'explorer ces chemins si aucune alternative nettement plus avantageuse n'existe.

L'application de ces poids aux connexions intra-étage et inter-étages permet d'adapter le comportement du robot en fonction de la complexité de la navigation. Par exemple, les transitions entre étages ont des poids légèrement plus élevés que celles effectuées au sein d'un même étage, ce qui reflète le coût supplémentaire associé aux déplacements verticaux (escaliers, ascenseurs, etc.).

Grâce à cette pondération adaptative, IntelliNav optimise les choix de trajectoire en fonction des conditions dynamiques de l'environnement, garantissant ainsi une navigation efficace et robuste tout en réduisant le risque d'impasses.

État	**Poids Intra-Floor**	**Poids Inter-Floor**	**Correspondance**
État 0	2	4	Non testé
État 1	1	3	Validé
État 2	10	12	Potentiellement bloqué
État 3	999	999	Bloqué

TABLE 3.1 – Association des états des arêtes et des poids

Le tableau 3.2 présente les transitions d'état possibles pour les arêtes du graphe HSTM, en précisant les événements déclencheurs et leurs effets sur la navigation.

3.3.5.2 Mise à jour des attributs des nœuds

Les nœuds de la HSTM, qu'ils représentent des objets (N_{obj}) ou des pièces (N_{room}), sont également ajustés en fonction des observations faites par le robot, lors de la navigation locale. Ces informations sont mises à jour en temps réel pour garantir une cohérence entre la carte et l'environnement réel.

— Objets non trouvés à leur position attendue (Cas des objets déplacés ou absents) : lorsque le robot atteint une pièce où un objet est censé être présent mais qu'il ne le détecte pas, le poids du nœud correspondant dans la HSTM est mis à jour pour refléter

Transition	**Déclencheur / Condition**	**Effet sur la navigation / Remarques**
État 0 → État 1 (Non testé → Validé)	Le robot découvre et traverse un chemin pour la première fois	Le chemin est désormais considéré fiable ; il est priorisé dans les futures planifications.
État 0 → État 3 (Non testé → Bloqué)	Un obstacle permanent est détecté sur un chemin encore jamais emprunté	L'arête est exclue temporairement des trajets planifiés.
État 1 → État 3 (Validé → Bloqué)	Un obstacle permanent est détecté sur un chemin précédemment utilisé	L'arête est bloquée temporairement pour le reste de la mission.
État 3 → État 2 (Bloqué → Potentiellement bloqué)	En fin de mission, les chemins bloqués sont requalifiés comme incertains	Mémorise l'historique ; rend l'arête réutilisable si aucune autre option n'est disponible.
État 2 → État 1 (Potentiellement bloqué → Validé)	Le robot réutilise le chemin avec succès	L'arête est de nouveau fiable ; son poids est abaissé.
État 2 → État 3 (Potentiellement bloqué → Bloqué)	Un nouveau blocage est confirmé par observation	Le chemin est à nouveau bloqué. L'arête est exclue temporairement des trajets planifiés.

Table 3.2 – Transitions d'états des arêtes dans IntelliNav

l'absence de l'objet. Cette mise à jour empêche le robot d'essayer de trouver un objet à un emplacement où il n'existe plus, évitant ainsi des déplacements inutiles. Si l'objet est localisé dans une autre pièce grâce au mode d'exploration sémantique, le robot met à jour les coordonnées du nœud correspondant. Cette mise à jour permet d'ajuster les futurs plans de navigation afin de rejoindre la nouvelle position de l'objet.

— Stratégie multi-cibles pour les objets : lorsqu'un objet cible peut se trouver à plusieurs emplacements, le robot explore successivement les nœuds correspondants. Si l'objet est localisé, la mission est considérée comme réussie. Si l'objet n'est pas trouvé dans un emplacement, ce nœud est marqué comme exploré et son poids est augmenté afin de le revisiter le moins souvent possible lors de la mission en cours. Le processus se poursuit jusqu'à ce qu'un objet déjà cartographié soit trouvé ou que toutes les alternatives soient épuisées, entraînant l'échec de la mission.

Toutefois, cette modification des poids reste temporaire. Une fois la mission terminée, qu'elle soit un succès ou un échec, tous les nœuds marqués comme explorés retrouvent leur état initial. Ce mécanisme garantit que les ajustements effectués pendant l'exploration n'altèrent pas définitivement la structure de la carte sémantique et que la navigation reste adaptative et cohérente sur le long terme.

L'intégration de mises à jour dynamiques dans IntelliNav présente de nombreux avantages et joue un rôle central dans l'efficacité des trajectoires planifiées. Elle favorise une adaptation immédiate aux changements environnementaux et améliore la fiabilité du système pour des missions futures en intégrant une mémoire persistante des obstacles et des difficultés rencontrées. IntelliNav garantit que la navigation repose sur des données à jour, minimisant ainsi le risque d'échecs liés à des obstacles imprévus. Grâce à la mise à jour des arêtes et nœuds, le robot peut ajuster son itinéraire face aux changements dans la connectivité des pièces ou aux déplacements d'objets et favoriser les trajets les plus sûrs et efficaces, en excluant les chemins bloqués ou en adaptant les coûts des connexions inter-pièces. Contrairement aux approches épisodiques, IntelliNav conserve une mémoire des obstacles rencontrés, améliorant ainsi les performances des navigations futures. Le robot évite ainsi les chemins à risque et privilégie les routes jugées fiables.

Le mécanisme de mémoire à long terme dans IntelliNav repose sur un changement des valeurs des états dans la carte HSTM, plutôt que sur l'accumulation de nouvelles informations. À chaque mission, les états des arêtes et des nœuds sont mis à jour pour refléter les conditions actuelles de l'environnement, basées sur les observations effectuées par le robot. Ces états, tels que "bloqué", "validé" ou "potentiellement bloqué", sont des valeurs sémantiques prédéfinies et reflètent directement les éléments rencontrés lors de la navigation.

Ainsi, au lieu d'ajouter de nouvelles données à chaque mission, IntelliNav modifie les valeurs de ces états en fonction des situations observées. Par exemple, lorsqu'un chemin est bloqué, l'état de l'arête correspondante passe à "bloqué", et si un blocage se produit de manière répétée, un compteur d'incidents est incrémenté. Ce compteur sert d'alerte pour signaler des chemins nécessitant une vérification supplémentaire, sans pour autant accumuler de données supplémentaires ou obsolètes. Lorsqu'un chemin devient à nouveau "validé", ce compteur est réinitialisé, assurant que seule l'information pertinente et récurrente est conservée dans la mémoire.

Ce système de mise à jour veille à ce que la carte reflète de manière dynamique les changements de l'environnement, tout en évitant l'accumulation d'informations non pertinentes. Ainsi, la mémoire d'IntelliNav reste concise et en phase avec les conditions réelles de navigation. Cette stratégie optimise la gestion de la mémoire. Elle améliore également les performances de navigation à long terme.

3.3.5.3 Gestion des blocages successifs

Pour mieux gérer la navigation à long terme et améliorer la planification de trajectoire, un mécanisme de mémoire est intégré afin de suivre le nombre de fois qu'une arête a été bloquée consécutivement. Lorsqu'une arête passe de l'état « potentiellement bloquée » à « bloquée » lors de tentatives successives, un compteur est incrémenté. Si cette arête reste bloquée après un nombre défini de blocages consécutifs (par exemple, 3 ou 4, selon la configuration de l'utilisateur), elle restera dans l'état « bloqué » même après la fin de la mission. Cela permet au robot de signaler ce chemin comme problématique, incitant l'utilisateur à inspecter l'itinéraire

pour identifier la nature de l'obstacle (ex. obstruction temporaire ou obstacle tendant à devenir permanent). Ce mécanisme de mémoire permet au robot de s'adapter à des situations où certains chemins sont fréquemment bloqués.

Si l'arête redevient valide (i.e. le chemin est à nouveau libre), le compteur de blocages est réinitialisé à zéro, assurant que les blocages répétés soient correctement cartographiés. Cela évite l'accumulation d'informations non pertinentes et garantit que la mémoire à long terme reflète uniquement les blocages persistants et récurrents.

Le mécanisme de mémoire à long terme d'IntelliNav ne consiste pas à accumuler des données au fil du temps. Il repose plutôt sur la mise à jour des états dans la HSTM afin de refléter les conditions environnementales les plus pertinentes et actuelles rencontrées durant la navigation. Le système modifie l'état de chaque arête et nœud en fonction des observations en temps réel, garantissant que la carte reflète l'état le plus à jour à la fin de chaque mission. Cette approche évite l'accumulation de données obsolètes ou redondantes : seuls les états sont mis à jour, sans ajout de nouvelles entrées. Ainsi, la mémoire reflète fidèlement les changements environnementaux réellement rencontrés, assurant que la carte reste précise sans croître inutilement au fil du temps.

3.3.6 Replanification du chemin global sémantique

La replanification du chemin sémantique global est importante dans le framework IntelliNav. Elle traite efficacement les situations où un chemin initialement prévu devient inaccessible en raison de changements imprévus dans l'environnement. Ce mécanisme vise à garantir que le robot puisse poursuivre sa mission en explorant des alternatives viables, au lieu de terminer prématurément avec un état d'échec comme illustré sur la figure 23. Inspiré du comportement humain, où un individu chercherait un autre chemin pour atteindre son objectif, la replanification sémantique globale donne au robot une capacité d'adaptation essentielle dans des environnements dynamiques.

3.3.6.1 Blocage de chemin : définition

Un obstacle rencontré sur le chemin global peut être classé comme :

— Obstacle contournable : Défini comme un obstacle local qui peut être évité en ajustant la trajectoire lors de la navigation locale sur la costmap (Layer 1 de la HSTM). Ces obstacles ne bloquent pas entièrement le chemin global, mais nécessitent des ajustements locaux pour les contourner. Par exemple, une chaise ou un petit objet bloquant un passage peut être contourné par le robot en recalculant une trajectoire locale pour atteindre le sous-objectif tout en restant sur le chemin global.

— Obstacle bloquant : Si plusieurs tentatives d'évitement échouent lors de la planification locale, l'obstacle est alors défini comme persistant. Il empêche l'accès au prochain sous-objectif. Des exemples incluent une porte fermée, un obstacle volumineux bloquant totalement l'accès (comme illustré sur la figure 23), ou une transition inter-étages inac-

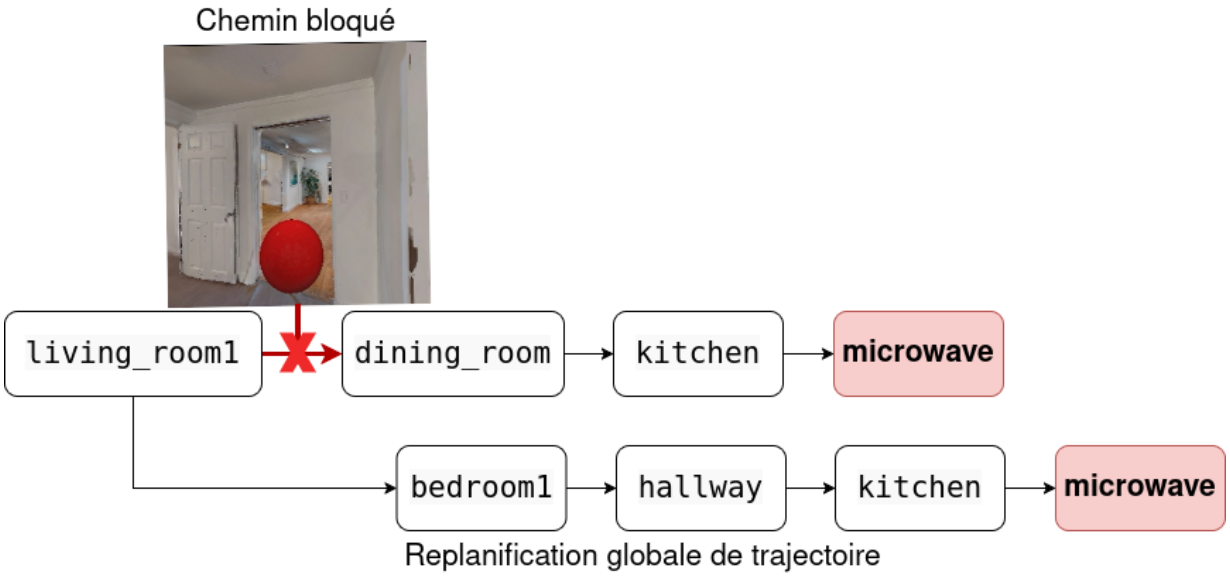


FIGURE 23 – Replanification du chemin global sémantique

cessible (ex. escalier). Dans ce cas, le chemin global est considéré comme bloqué. Aucune trajectoire locale alternative ne peut être générée pour contourner l'obstacle tout en respectant le chemin global, et une replanification globale sémantique devient nécessaire. Un exemple illustratif est présenté dans la figure 24, où un objet (un ballon rouge) bloque le passage à travers une porte.

Pour gérer ces cas, un mécanisme de vérification est appliqué afin de déterminer si une replanification globale est nécessaire. Comme illustré dans la figure 23, le chemin global planifié va du salon 1 jusqu'à l'objet micro-ondes. Lors de la navigation, ce chemin est découpé en sous-objectifs successifs (par exemple salon 1 → salle à manger → cuisine → micro-ondes). À chaque transition entre deux sous-objectifs, le robot vérifie si la trajectoire locale générée sur la costmap reste cohérente avec les pièces concernées. Pour ce faire, la trajectoire locale générée sur la costmap est échantillonnée et comparée aux polygones représentant les deux pièces concernées (pièce d'origine et pièce de destination).

Si tous les points de la trajectoire appartiennent exclusivement à ces deux pièces, cela signifie que, malgré la présence éventuelle d'un obstacle, le contournement a pu être effectué localement sans quitter l'espace des sous-objectifs. Dans ce cas, la navigation se poursuit normalement sans déclencher de replanification globale.

En revanche, si certains points de la trajectoire régénérée appartiennent à d'autres pièces de la carte, cela indique que le robot tente de contourner l'obstacle en sortant de l'espace défini par les deux sous-objectifs. L'arête reliant ces pièces est alors considérée comme bloquée et une replanification globale sémantique est déclenchée depuis la position courante du robot, afin de trouver un nouveau chemin évitant cette connexion.

Cette approche garantit que le robot peut adapter sa navigation même dans des environnements dynamiques, où des obstacles peuvent bloquer de manière inattendue les chemins planifiés. La capacité de mettre à jour la HSTM et de replanifier la trajectoire en fonction des



(a) Vue RGB perçue par le robot. (b) Image de profondeur. (c) Vue top-down de la scène.

FIGURE 24 – Exemple de visualisation d'un obstacle sur une porte dans une scene du dataset HM3DSem. Le ballon rouge représente l'obstacle.

changements en temps réel permet à IntelliNav de maintenir une navigation fiable dans des environnements sujets à modification.

3.3.6.2 Processus de replanification

Lorsqu'un obstacle bloquant est détecté, le système déclenche un processus de replanification global, structuré en plusieurs étapes clés :

— Mise à jour de la carte HSTM : Lorsque la nécessité d'un recalcul est détectée, le premier réflexe du système est de mettre à jour la HSTM pour refléter l'état actuel de l'environnement présenté à la section 3.3.5

— Recalcul du chemin global : Une fois la carte mise à jour, le système procède au recalcul du chemin sémantique global. Ce recalcul repose sur le même algorithme de Dijkstra que pour la planification globale initiale, qui prend en compte les poids mis à jour des arêtes, les nouvelles informations environnementales, pour refléter avec précision les conditions actuelles de navigation. Le nœud actuel du robot $N_{current}$ devient le nouveau point de départ du recalcul et l'algorithme cherche le chemin optimal vers l'objectif final N_g.

Si un nouveau chemin viable est trouvé, le robot reprend sa navigation en suivant les nouveaux sous-objectifs définis par cette nouvelle trajectoire. En revanche, si aucune alternative n'est possible (par exemple, si toutes les routes menant à l'objectif sont bloquées), la mission est déclarée échouée, et le robot passe à l'état *Failed*.

La replanification du chemin global sémantique dans IntelliNav renforce l'adaptabilité du système. Il donne au robot la capacité de contourner des situations imprévues, ce qui réduit les risques d'échec de mission. Inspiré du comportement humain, le système intègre un raisonnement sémantique et spatial pour ajuster la trajectoire et identifier des alternatives viables. De plus, l'utilisation des ressources, grâce à l'intégration des données historiques et des mises à jour en temps réel, minimise les efforts inutiles tout en maximisant les chances de succès. Ce mécanisme, en distinguant les obstacles contournables des obstacles persistants et en exploi-

tant les mises à jour dynamiques de la HSTM, constitue une solution scientifique et technique, garantissant une navigation fiable et adaptative.

3.4 Conclusion

Le framework IntelliNav constitue une avancée dans la navigation sémantique en milieu intérieur dynamique. Il repose sur une HSTM pour permettre une navigation explicable, adaptative et fiable. L'approche adoptée dans IntelliNav combine planification globale et navigation locale. Cette combinaison assure une progression fluide du robot et une adaptation efficace aux changements imprévus de l'environnement. Grâce à une segmentation claire des états du robot, le système assure une gestion transparente des missions, où chaque transition d'état est motivée par des observations et des mises à jour en temps réel. L'un des éléments centraux d'IntelliNav réside dans sa capacité à effectuer des mises à jour dynamiques de la carte HSTM, en intégrant des informations sur la position des objets, la connectivité des pièces et l'accessibilité des chemins. Cette mise à jour rend possible une planification de chemin évolutive, où chaque parcours est ajusté en fonction des nouvelles contraintes environnementales.

Par ailleurs, le mécanisme de replanification global renforce l'adaptabilité du système en offrant une alternative viable lorsque le chemin initialement prévu devient impraticable. En intégrant un raisonnement sémantique et spatial avancé, IntelliNav optimise chaque recalcul de chemin, minimisant les interruptions et maximisant l'efficacité de la navigation. Grâce à l'intégration de mémoires persistantes, le système apprend de ses expériences passées et améliore progressivement ses performances, assurant une navigation de plus en plus fiable au fil du temps.

Enfin, l'explicabilité inhérente à IntelliNav en fait une solution particulièrement adaptée aux applications réelles, notamment dans des contextes industriels où la transparence des décisions est essentielle.

L'ensemble de ces mécanismes positionne IntelliNav comme un framework efficace pour la navigation robotique sémantique, capable d'opérer dans des environnements complexes où les configurations spatiales et les objets peuvent évoluer dynamiquement. Cette approche hybride constitue une réponse pertinente aux défis posés par la navigation autonome en environnements dynamiques.

CHAPITRE 4

Expérimentation et évaluation

Ce chapitre présente l'évaluation complète des frameworks IntelliMap et IntelliNav présentés dans les chapitres précédents, conçus pour la navigation sémantique dans des environnements intérieurs. L'expérimentation est structurée en deux parties principales :

- Evaluation d'IntelliMap, qui analyse la précision et l'efficacité du processus de cartographie sémantique dans des environnements intérieurs domestiques et de bureau.
- Evaluation d'IntelliNav, qui examine les performances du système de navigation sémantique selon plusieurs critères, tels que la performance computationnelle de la planification globale, l'effet de la mise à jour dynamique de la carte sur la replanification, l'impact de la validation des sous-objectifs de la navigation sémantique.

4.1 Protocole expérimental

Les expérimentations menées visent à valider les capacités du framework IntelliNav et du module de cartographie IntelliMap dans des environnements intérieurs complexes et dynamiques. Pour cela, des benchmarks ont été utilisés, notamment le dataset HM3DSem [76], qui fournit des reconstructions réalistes de scènes intérieures annotées sémantiquement. Ce dataset sert à évaluer la performance de la navigation dans des environnements variés, comme des appartements résidentiels. Les environnements sélectionnés pour les tests incluent différentes configurations spatiales afin d'évaluer les performances du système dans divers contextes. Parmi ces environnements, on trouve les scènes 00824 et 00829, qui représentent des appartements à un étage comprenant une diversité de pièces et d'objets (respectivement 175 et 104 objets), ainsi que les scènes 00843 et 00862, qui sont des environnements multi-étages complexes intégrant des connexions inter-étages et contenant entre 145 et 388 objets. Ces scènes offrent un cadre d'évaluation pour tester la robustesse du système face à des défis tels que la navigation inter-étages,

la gestion d'un grand nombre d'objets et l'adaptabilité aux changements dynamiques.

Les tests ont été réalisés sur un système performant équipé d'un processeur Intel Core i9, de 64 Go de RAM et d'une carte graphique NVIDIA GeForce RTX 3000. Cette configuration matérielle assure l'exécution des algorithmes en temps réel tout en garantissant une gestion efficace des mises à jour sémantiques et des plans de navigation.

Pour compléter ces évaluations, deux autres types de tests ont été menés :

— Simulation avec le Dataset uHumans2 : Une simulation préliminaire a été réalisée en utilisant le dataset uHumans2, un environnement photo-réaliste généré avec Unity. Ce simulateur reproduit un environnement de bureau complexe, comprenant des objets variés (bureaux, chaises, salles de réunion) et des espaces structurés (couloirs, pièces). Cette simulation a permis de valider l'efficacité d'IntelliMap dans des environnements virtuels complexes, en mettant en avant sa capacité à générer des cartes sémantiques hiérarchiques précises et à faciliter la navigation sémantique.

— Tests en Environnement Réel avec le Robot Scout pour la cartographie : En complément des simulations, des tests ont été réalisés en environnement réel à l'aide du robot Scout, un véhicule autonome équipé d'un capteur LiDAR Hesai Pandar XT32 et d'une caméra Intel RealSense D455. Ces capteurs permettent une acquisition riche et précise des données environnementales, essentielle pour la navigation sémantique dans des contextes industriels. Ces tests ont permis de valider l'efficacité d'IntelliMap dans des environnements industriels réels, où la précision et la robustesse sont critiques pour des applications telles que la logistique autonome.

Ces expérimentations, combinant simulations et tests en environnement réel, offrent une évaluation complète des performances d'IntelliMap et IntelliNav.

4.2 IntelliMap : résultats des expérimentations

Pour évaluer l'efficacité et les performances d'IntelliMap, plusieurs expérimentations ont été conduites. Ces évaluations couvrent des aspects clés du framework, notamment la précision de la cartographie et l'efficacité mémoire. Voici un aperçu structuré des résultats obtenus.

4.2.1 Précision de la cartogaphie

L'évaluation de la précision de la cartographie vise à tester la capacité d'IntelliMap à localiser précisément les objets dans un environnement intérieur donné. Cette précision est essentielle pour garantir que le robot peut interagir de manière fiable avec son environnement, en particulier dans des tâches critiques comme la localisation d'objects. Pour évaluer cette précision, des tests ont été menés en mesurant la distance entre les positions réelles des objets (ground truth) et les positions calculées par IntelliMap. Un exemple illustrant à la fois la détection des objets à partir de la vue RGB et leur projection 3D dans le repère de la caméra est présenté en figure 25. Pour mesurer la précision de la cartographie, une méthodologie a été mise en

place. Les tests ont été réalisés en conditions réelles dans deux environnements distincts : un bureau et un appartement, tous deux meublés principalement avec des objets mobiles tels que des chaises et des tables. Pour l'évaluation, nous avons utilisé un ensemble contrôlé composé de deux tables, cinq chaises, un téléviseur et deux coussins. Ces objets ont été déplacés à plusieurs distances dans les deux environnements afin de comparer la précision de localisation. Chaque test a été répété cinq fois par objet, ce qui permet d'obtenir une moyenne représentative. Les distances choisies étaient de 1 mètre, 1,5 mètres, et 2 mètres par rapport à une référence fixe.

Le processus d'évaluation a suivi les étapes suivantes :

— Acquisition des données : Le robot Scout, équipé d'une caméra Intel RealSense D455, a été utilisé pour capturer les données environnementales. Ces capteurs permettent une acquisition des positions des objets dans l'espace.

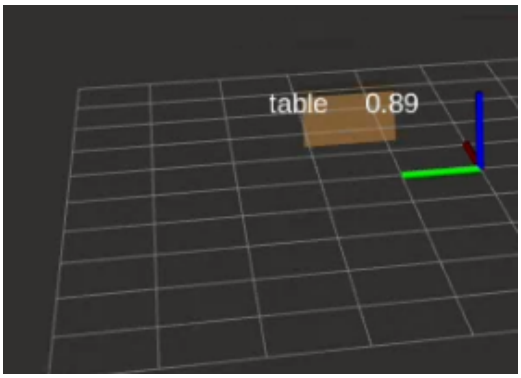
— Génération de la carte sémantique : Les données capturées ont été traitées par le module de cartographie des objets d'IntelliMap pour générer une carte des objets de l'environnement.

— Comparaison avec la vérité terrain (Ground Truth) : Les positions des objets calculées par IntelliMap ont été comparées aux positions réelles (ground truth) mesurées manuellement. Cette comparaison a permis de calculer l'erreur de localisation pour chaque objet, puis une moyenne a été établie pour évaluer la précision globale du système.

Les résultats des tests montrent qu'IntelliMap atteint une précision remarquable dans la localisation des objets. L'erreur moyenne de localisation est de 1,43%, ce qui signifie que les positions calculées par IntelliMap sont précises à quelques centimètres près des positions réelles. Par exemple pour un objet placé à 1 mètre, la position calculée par IntelliMap était en moyenne à 1,014 mètres.



(a) Image RGB capturée par la caméra embarquée, avec les boîtes englobantes des objets détectés par Omni3D projetées sur l'image.



(b) Projection 3D des objets détectés dans le repère de la caméra, permettant le calcul précis de leur position relative.

FIGURE 25 – Illustration de l'évaluation de la précision du mapping avec IntelliMap à partir de détections fournies par Omni3D.

4.2.2 Analyse de l'empreinte mémoire du stockage de la HSTM

L'empreinte mémoire est un critère essentiel pour les systèmes de cartographie sémantique, en particulier dans des environnements complexes et dynamiques où les ressources computationnelles sont limitées.

L'objectif de cette expérimentation est d'évaluer la mémoire nécessaire pour stocker les cartes générées par la HSTM, en comparaison avec d'autres frameworks de cartographie sémantique, tels que Hydra [14], VLMaps [77], et ConceptGraphs [78]. Deux aspects principaux ont été analysés pour cette évaluation. Premièrement, une comparaison complète des cartes a été réalisée, incluant toutes les couches de la carte, c'est-à-dire la couche métrique ainsi que les couches sémantiques. Deuxièmement, une comparaison des couches non métriques a été effectuée, se concentrant uniquement sur les couches sémantiques et excluant la couche métrique, afin de mesurer l'efficacité de la représentation sémantique indépendamment de la représentation métrique. Pour cette analyse, deux datasets ont été utilisés. Le premier, uHumans2 Office, représente un environnement à un seul étage simulant un bureau avec des pièces fonctionnelles distinctes. Il sert à évaluer la performance de la HSTM dans un contexte simple. Le second, HM3DSem, est un ensemble de scènes multi-étages, offrant un cadre plus complexe pour tester la robustesse et l'efficacité de la HSTM dans des environnements hiérarchiques et étendus.

4.2.2.1 Résultats sur le dataset "uHumans2 Office"

Le dataset uHumans2 Office a été utilisé pour comparer la consommation mémoire de la HSTM avec celle du framework Hydra. Les résultats illustrés sur le tableau 4.1 montrent une réduction significative de la mémoire nécessaire pour IntelliMap :

— Comparaison complète des cartes : IntelliMap (avec sa couche métrique sous forme de costmap) consomme seulement 16,1 Mo. Hydra (utilisant une représentation par maillage (mesh) pour la couche métrique) nécessite 164,1 Mo. Cela représente une réduction de 90,2% pour IntelliMap.

— Comparaison des couches non métriques : IntelliMap consomme 0,156 Mo pour les couches sémantiques. Hydra nécessite 32,5 Mo pour les mêmes couches. IntelliMap utilise ainsi 325 fois moins de mémoire que Hydra pour les couches non métriques.

Au-delà de la compression métrique, IntelliMap se distingue dans cette évaluation, par une représentation sémantique hiérarchique organisée en trois couches complémentaires : costmap, des objets et des pièces. Cette structuration permet de représenter fidèlement l'environnement tout en maintenant une compacité élevée. En comparaison, Hydra repose sur une représentation à cinq couches, avec une couche "places" très détaillée mais redondante pour la navigation, ce qui impacte aussi l'empreinte mémoire. Ces résultats démontrent que IntelliMap parvient à encoder les informations géométriques et sémantiques essentielles à la navigation avec une expressivité équivalente mais une empreinte mémoire bien plus faible, ce qui en fait un choix adapté aux systèmes embarqués ou aux environnements contraints en ressources. La représentation compacte de la HSTM, en particulier l'utilisation d'une costmap pour la couche métrique,

montre une réduction de la mémoire nécessaire, tout en conservant une précision élevée pour la navigation.

Environnement	**IntelliMap**		**Hydra**	
	Type carte	**MU (Mo)**	**Type carte**	**MU (Mo)**
uHumans2 Office (avec carte métrique)	Costmap	16.1	Mesh	164.1
uHumans2 Office (sans carte métrique)	Aucune carte	0.156	Aucune carte	32.5

TABLE 4.1 – Comparaison de l'utilisation mémoire (MU) avec et sans carte métrique. IntelliMap utilise une costmap comme carte métrique, tandis qu'Hydra utilise un maillage (mesh).

4.2.2.2 Résultats sur le dataset HM3DSem

Le dataset HM3DSem a été utilisé pour évaluer la performance de la HSTM dans des environnements multi-étages. Les résultats confirment l'efficacité mémoire de la HSTM. Le tableau 4.2 présente une comparaison des exigences de stockage mémoire pour différentes techniques de cartographie sémantique, mesurées en mégaoctets (Mo), sur plusieurs scènes du dataset HM3DSem.

— Par exemple sur la scène 00829 (multi-étages) : IntelliMap nécessite seulement 6,2 MB pour stocker toutes les couches de la HSTM (y compris la costmap et les couches sémantiques). En comparaison, d'autres techniques comme VLMaps (407 MB) et ConceptGraphs (110 MB) nécessitent des volumes de stockage bien supérieurs.

— Représentation compacte : La HSTM ne stocke pas de nuages de points, de maillages 3D, d'images ou de caractéristiques de haute dimension, ce qui contribue à sa compacité. Dans les environnements multi-étages, la mémoire totale est calculée en utilisant le stockage nécessaire pour tous les étages.

La HSTM montre une réduction significative de la mémoire nécessaire par rapport aux autres méthodes, en particulier dans des environnements multi-étages. Cette efficacité est due en partie à l'absence de stockage de données volumineuses (nuages de points, maillages, etc.) et à l'utilisation d'une représentation hiérarchique compacte.

La réduction de la mémoire nécessaire rend la HSTM adaptée à des plates-formes robotiques avec des contraintes de calcul et de stockage. IntelliMap ne conserve que les relations sémantiques et les informations essentielles à la navigation.

Les expérimentations sur les environnements uHumans2 (bureaux) et HM3DSem (domestiques multi-étages) montrent une empreinte mémoire réduite de près de 90% en moyenne, avec une efficacité de navigation maintenue.

Au-delà de l'efficacité en termes de stockage, les choix de conception de la HSTM sont également motivés par des considérations pratiques liées à la navigation en réel dans des environnements dynamiques. La couche métrique est implémentée sous forme de costmap (carte des coûts), évitant les représentations denses tout en facilitant la mise à jour locale et la planifica-

tion réactive face aux obstacles. Elle permet une navigation de bas niveau avec des algorithmes standards, assurant un comportement adaptatif et réactif.

Il convient néanmoins de relativiser la comparaison des résultats mémoire : les méthodes comme VLMaps [77], ConceptGraphs [78] ou HOV-SG [79] ne fournissent pas les mêmes fonctionnalités. Elles intègrent des modèles vision-langage pour le raisonnement en vocabulaire ouvert, via une couche d'interprétation reposant sur des embeddings visuels ou des graphes de scène. Leurs structures diffèrent : HOV-SG encode objets, pièces et relations multi-étages, tandis que ConceptGraphs se limite aux objets et attributs.

La HSTM, intègre explicitement les couches objets, pièces et étages dans une structure unifiée, modifiable dynamiquement. Elle prend en charge les mises à jour en ligne et une mémoire persistante, permettant la réutilisation inter-mission. Elle autorise également la décomposition d'objectifs sémantiques en sous-objectifs locaux, avec replanification déclenchée en cas de blocage, grâce aux retours de la costmap. À l'inverse, les systèmes comme HOV-SG génèrent leur graphe de navigation à partir d'un graphe de Voronoï extrait sur les zones de l'espace libre durant la phase de cartographie. Ce graphe reste statique durant le déploiement et ne prend pas en compte les modifications de traversabilité causées par des obstacles dynamiques. Comme explicitement indiqué dans leur travail, HOV-SG suppose un environnement statique et ne prend pas en charge la navigation dynamique.

L'évaluation de l'empreinte mémoire est privilégiée ici, car le processus modulaire de cartographie rend les temps d'exécution fortement dépendants de l'implémentation et du matériel. À l'inverse, la mémoire utilisée reste un indicateur stable et pertinent, en particulier pour des déploiements embarqués à long terme. La HSTM est générée une fois, hors ligne, puis mise à jour de manière incrémentale durant la navigation. L'optimisation du stockage est donc essentielle pour permettre une mise à l'échelle efficace, avec des mises à jour rapides dans des environnements dynamiques.

Scène	**Floor**	**VLMaps** [77]	**ConceptGraphs** [78]	**HOV-SG** [79]	**HSTM**
00824	1	568 MB	143 MB	143 MB	8.8 MB
00829	1	407 MB	110 MB	99 MB	6.2 MB
00843	2	534 MB	143 MB	125 MB	7.3 MB
00862	3	1808 MB	474 MB	479	27 MB

TABLE 4.2 – Comparaison de l'empreinte mémoire pour le stockage des cartes.

4.3 IntelliNav : résultats des expérimentations

4.3.1 Optimisation de la navigation par validation sémantique des pièces

L'optimisation de la navigation ne repose pas uniquement sur la génération d'un chemin global optimal, mais aussi sur l'efficacité de la gestion des sous-objectifs. Les tests ont été réalisés sur plusieurs missions en environnement intérieur sur les scènes 00824 et 00829. Pour chaque mission, la distance initiale (pré-calculée à partir des centres des pièces) a été comparée avec la distance réellement parcourue par le robot et la réduction du parcours grâce à la validation des pièces basée sur la localisation sémantique a été mesurée.

Les résultats présentés sur le tableau 4.3 indiquent une réduction significative des distances parcourues, démontrant une navigation plus fluide et plus efficace :

— Mission 1 : Living Room → Microwave : Réduction de 35.73%
— Mission 2 : Entryway → Bed : Réduction de 29.59%
— Mission 3 : Bedroom3 → Sofa : Réduction de 28.02%
— Mission 4 : Living Room → Toilet : Réduction de 21.81%

En moyenne, les trajets ont été optimisés de 21% à 35% grâce à l'intégration de la validation des pièces. Ces résultats démontrent l'importance de cette approche dans l'amélioration de l'efficacité de la navigation locale.

Scene	Mission	Distance initiale jusqu'à l'objectif (m)	Distance parcourue (m)	Réduction du trajet (%)
00824	living_room2 to bathtub	30.82	24.24	21.34
00824	living_room1 to microwave	18.86	12.12	35.73
00824	bedroom3 to sofa	22.09	15.90	28.02
00829	living_room to toilet	9.49	7.42	21.81
00829	entryway to bed	15.11	10.64	29.59

TABLE 4.3 – Comparaison de la longueur des trajectoires et de la réduction

4.3.2 Exploration et découverte d'objets non cartographiés

Dans des environnements intérieurs, certains objets peuvent ne pas être enregistrés dans la carte sémantique initiale, ce qui nécessite une stratégie d'exploration adaptative pour localiser efficacement ces objets. Pour répondre à ce défi, IntelliNav intègre une approche d'exploration sémantique qui permet au robot d'identifier les emplacements les plus probables où un objet recherché pourrait se trouver. Cette stratégie repose sur l'analyse des attributs sémantiques des pièces connues dans la HSTM, combinée à un modèle probabiliste pour estimer les zones les plus pertinentes à explorer.

Par exemple, pour localiser une tasse dans un environnement composé des pièces suivantes : [cuisine, salon, salle de bain, chambre1, chambre2, bureau], le prompt suivant a été utilisé :

```
J'ai un environnement modélisé sous forme de graphe sémantique avec les pièces
suivantes : [cuisine, salon, salle de bain, chambre1, chambre2, bureau].
Je cherche une 'tasse'. Donne une liste des pièces avec une estimation
de probabilité de présence en pourcentage.

Retourne uniquement les pièces avec une probabilité supérieure à 40% sous forme
de liste [(pièce, probabilité)].
```

Les résultats retournés par les LLMs sont structurés sous forme de listes de tuples [(pièce, probabilité)], ce qui permet une intégration directe dans le système de décision du robot. Par exemple, le modèle GPT-4o retourne :

```
[
["cuisine", 80],
["salon", 50]
]
```

tandis que Deepseek propose :

```
[
["cuisine", 80],
["bureau", 60],
["salon", 50]
]
```

Ces résultats mettent en évidence la cuisine comme l'emplacement le plus probable pour trouver une tasse, suivi du bureau et du salon. Cette approche probabiliste permet au robot de prioriser ses actions d'exploration en se concentrant sur les zones les plus pertinentes, améliorant ainsi l'efficacité de la recherche dans des environnements complexes et dynamiques.

4.3.3 Performance computationnelle de la planification sémantique globale des trajets

L'évaluation des performances computationnelles d'IntelliNav a été réalisée afin de mesurer l'efficacité du système en termes de génération de chemins sémantiques et d'optimisation des ressources de calcul. L'objectif principal de cette expérimentation était d'analyser le temps nécessaire pour générer un chemin global sémantique et d'évaluer le taux de succès des missions en fonction des objectifs assignés.

L'expérimentation a été menée dans un environnement connu, en utilisant la HSTM de la scène 00824 issue du dataset HM3DSem.

4.3.3.1 Évaluation de la planification vers des objectifs cartographiés

Pour garantir une évaluation représentative, un ensemble de 50 essais a été effectué, chaque essai consistant en une navigation entre une position de départ aléatoire et un objectif sémantique sélectionné directement depuis le graphe de scène. Deux types de missions ont été considérés : la navigation vers une pièce spécifique (40% des essais) et la navigation vers un objet cible (60% des essais).

Les chemins générés lors de ces essais variaient de 1 à 9 nœuds, selon la position de départ, l'objectif sémantique, et la structure du graphe. Lorsque l'objet cible se trouve dans la même pièce que la position de départ, le chemin contient uniquement 2 nœuds (par exemple, kitchen → table dans la même pièce). À l'inverse, des trajets plus longs sont requis lorsque des transitions inter-pièces ou inter-étages sont nécessaires. Sur la scène 00824, pour naviguer du bathroom2 au microwave situé dans la cuisine, le chemin s'étend sur 5 nœuds, impliquant une succession de pièces connectées. Sur la scène 00862 (multi-étagée), un trajet typique du kitchen au 1er étage vers bathroom3 au 3e étage comprend 6 nœuds, incluant deux transitions verticales par escalier.

Les résultats obtenus montrent que le taux de réussite a atteint 100%, confirmant que tous les objectifs étaient atteignables dans l'environnement testé grâce à la carte sémantique préalablement construite. Cette performance souligne l'efficacité de l'intégration des informations sémantiques dans la planification des chemins, qui guide le robot de manière cohérente et systématique vers ses cibles, sans rencontrer d'impasses logiques.

En ce qui concerne le temps de calcul, les mesures effectuées ont révélé une moyenne de 3,0 millisecondes par essai pour la génération d'un chemin global sémantique. Ce temps inclut l'extraction et l'identification des nœuds du graphe HSTM, ainsi que le calcul du chemin optimal vers la destination. Cette rapidité démontre la capacité du système à supporter des prises de décision quasi-instantanées, un atout crucial pour une navigation dans des environnements dynamiques où les conditions peuvent évoluer rapidement.

4.3.3.2 Évaluation de la planification vers des objets non cartographiés

Pour tester la robustesse d'IntelliNav dans de telles situations, une seconde série d'expériences a été menée en simulant la recherche de 50 objets non cartographiés.

Dans cette configuration, le système a exploité ChatGPT pour générer des hypothèses sur les emplacements les plus probables des objets en fonction des attributs sémantiques des pièces tels qu'enregistrés dans la HSTM. Cette approche combine la connaissance sémantique de l'environnement avec une stratégie de recherche guidée pour optimiser les déplacements du robot.

Les résultats montrent un taux de réussite de 55%, indiquant que dans plus de la moitié des cas, le robot a pu planifier une trajectoire vers la pièce estimée pour l'objet recherché, bien que celui-ci n'ait pas été initialement présent dans la carte sémantique. Le modèle LLM a estimé la localisation des objets dans des pièces de la HSTM, en générant des hypothèses sur les zones probables où les objets pouvaient se trouver, avec un seuil de confiance de 80%. Ce taux de

confiance reflète la fiabilité des hypothèses générées par le modèle.

En conclusion, les résultats obtenus mettent en évidence la robustesse et l'efficacité computationnelle d'IntelliNav, capable de générer rapidement des plans de navigation exploitables tout en garantissant un taux de réussite maximal pour les objectifs connus. L'intégration d'un mécanisme de recherche d'objets non cartographiés et l'adaptation des stratégies de navigation à des environnements partiellement inconnus constituent une avancée clé pour améliorer l'autonomie et la flexibilité des robots mobiles dans des scénarios réels.

L'une des forces du framework IntelliNav réside dans l'exploitation efficace des ressources computationnelles. L'approche sémantique utilisée ici structure l'information de manière compacte et hiérarchisée, réduisant ainsi les besoins en stockage et en puissance de calcul. Cette optimisation est particulièrement avantageuse pour les robots mobiles opérant avec des capacités matérielles limitées, garantissant un compromis entre précision et performance.

4.3.4 Optimisation dynamique des trajectoires par ajustement des états des connexions

L'optimisation des trajets et la gestion des connexions dans un environnement dynamique sont essentielles pour garantir une navigation efficace. IntelliNav adopte une approche adaptative où les connexions entre pièces sont pondérées dynamiquement en fonction de leur état. cette stratégie optimise la planification globale du robot tout en tenant compte des conditions environnementales. Cette section analyse l'impact de ces poids dynamiques, ainsi que les performances du système dans des scénarios à un ou plusieurs étages.

Scène	**Segment**	**Inter Floor**	**Poids initial**	**Poids modifié (A)**	**Poids modifié (B)**
00824	living_room1 → dining_room	no	2	1	1
	dining_room → kitchen	no	2	10	10
	hallway → kitchen	no	2	-	999
00843	hallway1 → bathroom2	no	2	10	10
	living_room → hallway1	no	2	-	10
00862	hallway1 (floor 1) → entryway (floor 2)	yes	4	999	10
	entryway (floor 2) → staircase (floor 3)	yes	4	10	10

TABLE 4.4 – Impact des modifications de l'état des bords sur la planification sémantique des chemins dans les scénarios à un étage et à plusieurs étages.

4.3.4.1 Expérimentation sur un environnement à un étage

Pour analyser l'impact de cette gestion dynamique, une série d'expériences a été réalisée sur la scène 00824 du dataset HM3DSem, représentant un appartement à un étage comprenant 10 pièces (salons, cuisine, chambres, salle de bains, etc.). L'objectif du robot est de naviguer de "living room1" jusqu'à un "microwave" situé dans la cuisine.

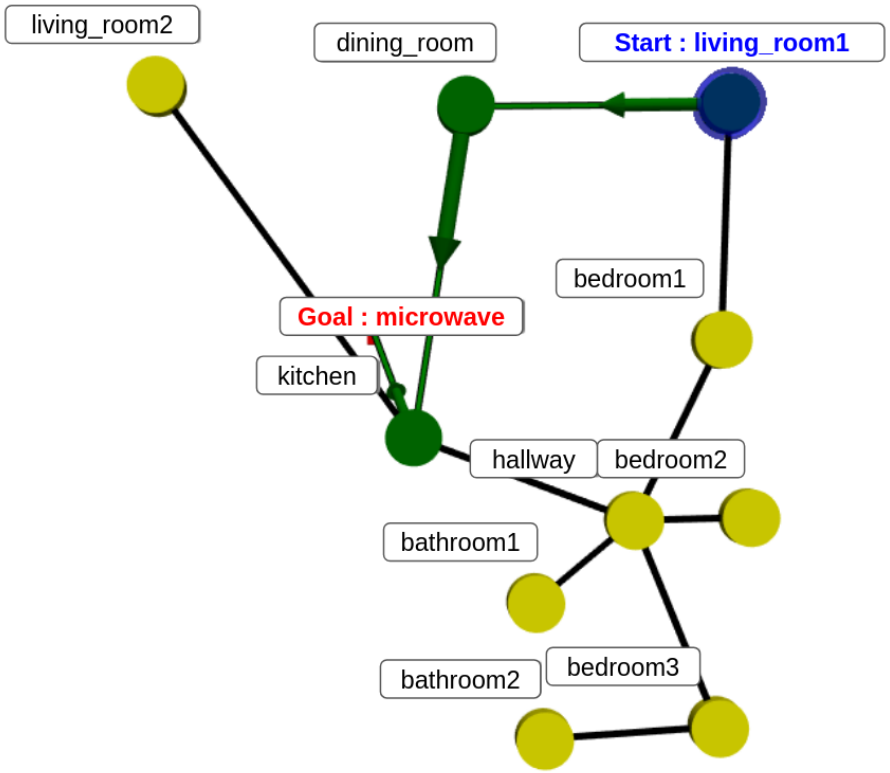
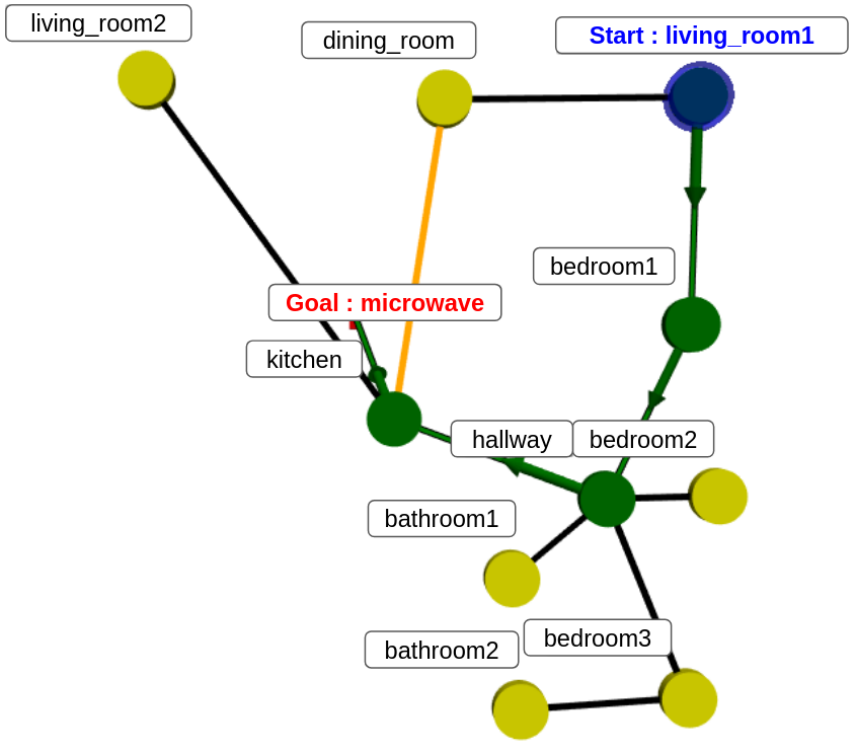


FIGURE 26 – Configuration par défaut : Trajet sémantique global de référence, de living_room1 à microwave, calculé avec les poids par défaut des arêtes de la HSTM de la scène 00824.

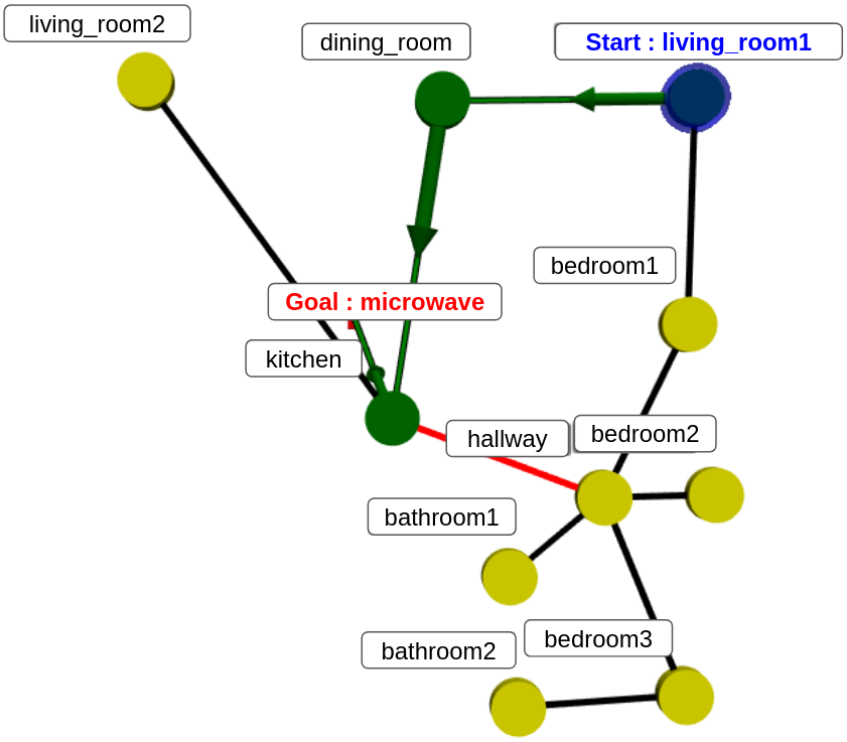
L'expérimentation a été menée selon trois configurations différentes dont les résultats sont présentés sur l'image 27 :

- (a) Configuration par défaut : Tous les chemins sont considérés comme "non testés" et ont des poids initiaux identiques.
- (b) Modification A : Certaines connexions sont marquées comme "potentiellement bloquées", notamment entre "dining room" et "kitchen".
- (c) Modification B : Une connexion est complètement bloquée entre "hallway" et "kitchen", obligeant le robot à contourner cet obstacle.

Dans le cas (a) présenté sur la figure 26, le robot a suivi le chemin le plus court, passant par "dining room" et "kitchen". L'objectif a été atteint en 23.639 secondes, avec une distance d'environ 0.5m par rapport au micro-ondes. Dans le cas (b) illustré sur la figure 27a, le poids de la connexion "dining room → kitchen" a été augmenté pour simuler une obstruction temporaire. Le robot a alors évité cette connexion et opté pour un chemin alternatif via "bedroom1" et "hallway", augmentant légèrement la distance parcourue. Dans le cas (c) illustré sur la figure 27b, la connexion entre "hallway" et "kitchen" a été complètement bloquée. Le robot a tenté plusieurs alternatives avant d'emprunter finalement la connexion potentiellement bloquée, démontrant sa capacité d'adaptation face aux contraintes. L'analyse des résultats montre que les connexions pondérées influencent significativement la planification globale sémantique des chemins. Elle orientent les décisions du robot de manière intelligente et adaptative, en favorisant l'évitement des zones problématiques.



(a) Modification A : Trajet avec des poids de nœuds augmentés. La connexion entre la salle à manger et la cuisine est considérée comme potentiellement bloquée (en orange), et la connexion entre le salon 1 et la salle à manger est déjà visitée.



(b) Modification B : Trajet avec des poids de nœuds augmentés. La connexion entre la salle à manger et la cuisine est considérée comme potentiellement bloquée, et la connexion entre le couloir et la cuisine est bloquée (en rouge).

FIGURE 27 – Comparaison des trajets sémantiques globaux avec des poids de nœuds variables. Les sous-figures illustrent l'impact de différentes configurations de poids sur le trajet sémantique global.

4.3.4.2 Expérimentation sur un environnement multi-étages

Afin d'évaluer la performance d'IntelliNav en scénario multi-niveaux, des tests ont été réalisés sur les scènes 00843 et 00862, comportant respectivement deux et trois étages. L'objectif est d'évaluer la gestion des transitions inter-étages, en particulier lorsque certaines connexions sont bloquées ou limitées. Dans ces scénarios, l'appartement en scène 00843 comprend 2 étages et 13 pièces. L'objectif du robot est de naviguer depuis un salon au premier étage vers une salle de bains. L'appartement en scène 00862 comprend 3 étages et 30 pièces. L'objectif du robot est d'atteindre une chambre au troisième étage depuis un couloir du premier étage.

— Priorisation des Trajets sur un même étage (scène 00843) : L'expérience démontre comment IntelliNav privilégie les trajets sur un même étage lorsque les transitions inter-étages entraînent des coûts de déplacement plus élevés. L'expérimentation a été réalisée sur la scène 00843, comme illustré dans la figure 28, qui comprend deux étages avec des distributions de pièces distinctes. L'étage 1 contient sept pièces : une buanderie, une cuisine, un salon, un couloir, deux salles de bain et une chambre. L'étage 2 contient six pièces : deux chambres, deux salles de bain, un placard et un couloir. Le robot commence dans le salon de l'étage 1 avec pour objectif d'atteindre une salle de bain. Quatre salles de bain sont accessibles depuis le salon, toutes situées à deux pièces de distance : *bathroom1* et *bathroom2* à l'étage 1, ainsi que *bathroom3* et *bathroom4* à l'étage 2. Dans la configuration de référence, IntelliNav sélectionne *bathroom2* à l'étage 1 comme destination du chemin global, privilégiant une salle de bain sur le même étage afin d'éviter le coût de transition entre étages. Le chemin choisi est :

$$\textit{living room} \rightarrow \textit{hallway1} \rightarrow \textit{bathroom2}$$

Ce choix illustre l'efficacité d'IntelliNav dans la sélection de l'option la plus proche et la moins coûteuse sur le même étage. Lorsque l'arête entre *hallway1* et *bathroom2* est modifiée en "potentiellement bloquée", IntelliNav met à jour son chemin global et sélectionne *bathroom1*, également situé à l'étage 1. Le nouveau chemin devient :

$$\textit{living room} \rightarrow \textit{hallway1} \rightarrow \textit{bathroom1}$$

Ce résultat met en évidence la capacité du système à s'adapter en privilégiant une autre option sur le même étage tout en évitant l'arête "potentiellement bloquée". Enfin, lorsque l'arête entre *living room* et *hallway1* est également marquée comme "potentiellement bloquée", IntelliNav ajuste son chemin global et sélectionne *bathroom3* à l'étage 2. Le chemin mis à jour est :

$$\textit{living room} \rightarrow \textit{hallway2} \rightarrow \textit{bathroom3}$$

Cette dernière modification illustre la capacité d'IntelliNav à gérer plusieurs contraintes et à prioriser les transitions inter-étages lorsqu'aucune option réalisable n'est disponible sur le même étage.

— Évaluation de la gestion des contraintes hiérarchiques dans un scénario à trois étages

Dans le scénario à trois étages, la capacité d'IntelliNav à gérer les contraintes hiérarchiques et à traiter l'inaccessibilité complète de certains étages est évaluée. L'expéri-

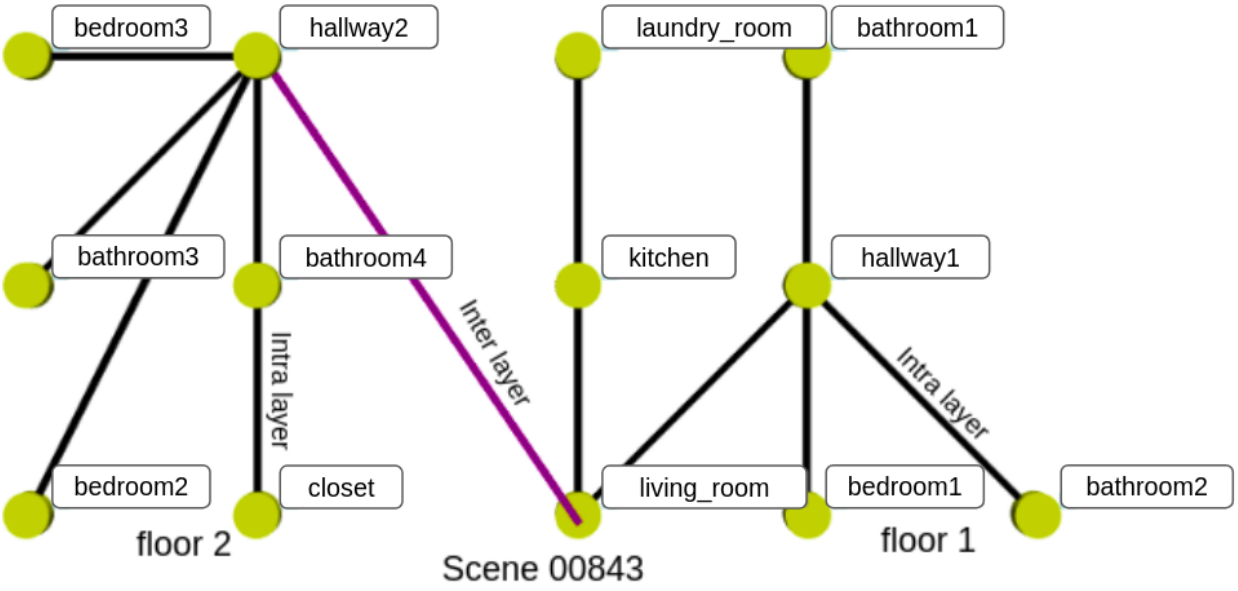


FIGURE 28 – Liste des pièces de la scène 00843 de la troisième couche de la HSTM

mentation a été réalisée sur la scène 00862, qui comprend un total de 30 pièces réparties sur trois niveaux, comme illustré dans la figure 29.

L'étage 1 contient 7 pièces, dont trois pièces non identifiées, un salon, une cuisine, une salle de bain et deux couloirs. L'étage 2 comporte 8 pièces, comprenant un couloir, une entrée, un placard, une salle de bain, un bureau, deux salons et une cuisine. L'étage 3 est le plus complexe, avec 15 pièces, incluant quatre placards, trois salles de bain, une pièce non identifiée, trois chambres, un couloir, un salon, un escalier et une buanderie. Le robot commence dans *hallway1* à l'étage 1, avec pour objectif d'atteindre une chambre située à l'étage 3. Lorsque l'arête inter-étages reliant l'étage 1 à l'étage 2 est marquée comme "bloquée" (Modification A), IntelliNav identifie les étages 2 et 3 comme entièrement inaccessibles, bien que l'étage 3 reste théoriquement navigable depuis l'étage 2. En conséquence, le robot restreint sa navigation à l'étage 1, échouant à générer un chemin global vers les chambres de l'étage 3. Ce comportement met en évidence la capacité d'IntelliNav à propager les effets d'une transition bloquée et à évaluer dynamiquement l'accessibilité des étages supérieurs en fonction des dépendances hiérarchiques.

Lorsque les arêtes inter-étages sont marquées comme "potentiellement bloquées" (Modification B), IntelliNav continue de générer un chemin global :

$$hallway1 \rightarrow entryway \rightarrow staircase \rightarrow hallway4 \rightarrow bedroom1$$

En tenant compte de la possibilité que ces transitions restent franchissables, le système exploite ces chemins incertains, illustrant son comportement adaptatif dans des scénarios avec des contraintes non définitives.

Les résultats des expérimentations présentés montrent que la gestion dynamique des connections entre pièces joue un rôle essentiel dans la capacité du robot à s'adapter aux environnements évolutifs. En intégrant une pondération adaptative des connexions, IntelliNav optimise

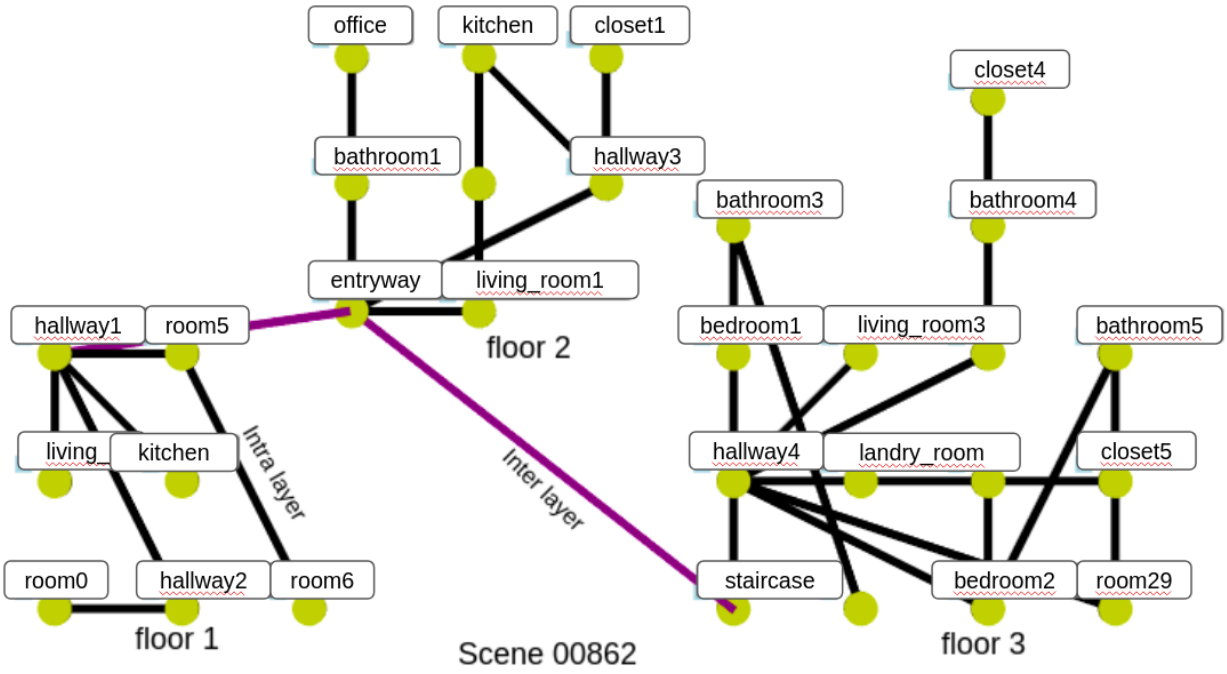


FIGURE 29 – Liste des pièces de la scène 00862 de la troisième couche de la HSTM

son processus de planification en fonction des conditions rencontrées.

Tout d'abord, cette approche permet une prise de décision contextuelle, essentielle pour garantir le succès des missions de navigation. En favorisant les connexions validées et en évitant autant que possible les chemins potentiellement bloqués, le robot minimise le risque d'échec. Cette stratégie assure que les trajets empruntés sont non seulement optimaux en termes de distance, mais également sécurisés en fonction des observations passées.

Dans les environnements multi-étages, la gestion des connexions prend une importance particulière. En attribuant un coût plus élevé aux transitions verticales, IntelliNav évite les déplacements inutiles entre les étages, favorisant la navigation sur un même niveau avant d'envisager un changement d'étage si aucune alternative viable n'est disponible.

Enfin, la robustesse face aux changements dynamiques constitue un atout majeur d'IntelliNav. Cette capacité d'adaptation garantit une navigation fluide et fiable, même dans des environnements où la disposition des pièces et la disponibilité des chemins peuvent évoluer de manière imprévisible.

En combinant ces différents éléments, IntelliNav se distingue par sa capacité à offrir une navigation sémantique intelligente, optimisée et résiliente face aux défis posés par les environnements intérieurs dynamiques et multi-niveaux.

4.3.5 Gestion des objets manquants ou dynamiques

Afin d'évaluer la robustesse de notre framework dans des scénarios dynamiques, nous avons simulé des situations dans lesquelles un objet, initialement cartographié lors de la phase d'exploration, n'est plus présent au moment de la navigation. Ces expériences permettent de tester

les mécanismes de mise à jour d'IntelliNav face à des incohérences au niveau des objets.

Deux types de scénarios représentatifs ont été considérés :

— **Objets à instance unique** : il s'agit d'objets présents en un seul exemplaire dans la carte, comme un réfrigérateur. Dans ce cas, l'objet est supprimé manuellement du maillage 3D de la scène, utilisé pour générer le layer 1 (représentation géométrique), tout en maintenant sa présence dans les couches sémantiques de la HSTM (layers 2 et 3). Lors de la navigation, le robot tente d'atteindre l'objet mais échoue à le détecter. S'il existe des alternatives dans la carte, le système peut alors basculer en mode d'exploration sémantique pour rechercher des instances non cartographiées de la même catégorie.

— **Objets à instances multiples** : ce cas concerne des objets tels que les lits ou les tables, présents en plusieurs exemplaires dans différentes pièces. Si la première instance ciblée est introuvable ou inaccessible, le planificateur tente dynamiquement d'accéder à d'autres instances connues. Le succès dépend alors de la capacité du système à réévaluer les trajectoires candidates et à mettre à jour sa représentation interne de manière cohérente.



(a) Maillage original avec l'objet *toilet* dans *bathroom1*.



(b) Maillage modifié simulant la disparition de l'objet *toilet*.

FIGURE 30 – Modification du maillage de la scène 00824 pour simuler la disparition dynamique de l'objet *toilet* dans *bathroom1*.

Pour simuler la disparition des objets, nous avons modifié les maillages de scènes extraits du jeu de données HM3DSem à l'aide de l'outil de modélisation 3D open-source Blender [80]. Plus précisément, la géométrie de l'objet est supprimée du maillage, tout en conservant son nœud associé dans la carte HSTM. La figure 30 illustre visuellement ce processus sur l'objet *toilet* dans la scène 00824. Bien que des artefacts résiduels (géométriques ou d'éclairage) puissent subsister à proximité de la zone modifiée, l'objectif est atteint : l'objet est devenu indétectable pour le modèle de détection utilisé durant la navigation, simulant ainsi efficacement sa disparition de l'environnement réel.

Cette méthodologie a été testée sur 10 objets différents : *toilet, table, bathtub, sofa, chair, refrigerator, bed, book, flower, picture*. Pour chaque objet, nous avons mesuré le taux de réussite de navigation, défini comme la capacité du robot à identifier et atteindre une instance alternative

de l'objet après avoir constaté son absence à l'emplacement attendu.

Sur les 10 objets testés, 8 cas ont abouti à une navigation réussie. Par exemple, pour l'objet *toilet*, celui-ci a été manuellement retiré de la salle de bain *bathroom1*. Lors de l'exécution, le robot a échoué à le détecter à cet emplacement, qui a alors été marqué comme visité. IntelliNav a ensuite redirigé la navigation vers une seconde instance cartographiée dans *bathroom2*, comme illustré dans la figure 31.



FIGURE 31 – L'objet *toilet* a été détecté avec succès dans *bathroom2*, confirmant la réussite de la mission.

En revanche, les expérimentations impliquant le *refrigerator* et la *bathtub* ont échoué. Ces objets ne disposaient que d'une seule instance connue dans la carte HSTM. Après l'échec de détection du réfrigérateur dans la cuisine d'origine, IntelliNav a déclenché un processus d'exploration. Le modèle GPT-4o a alors proposé des pièces alternatives, telles que la salle à manger (*dining_room*) avec un taux de confiance de 66%. Toutefois, aucune nouvelle instance n'a été détectée dans cette pièce. Une situation similaire s'est produite pour la baignoire, avec la pièce *bathroom2* suggérée à 90% de confiance, mais là encore sans succès. Dans ces deux cas, malgré la suggestion pertinente du LLM et les tentatives d'exploration, aucun objet n'a été retrouvé. Les nœuds correspondants ont donc été mis à jour pour refléter l'absence confirmée des objets, et les pièces déjà visitées ont été marquées afin d'éviter toute redondance dans les tentatives futures.

Ces résultats mettent en évidence la robustesse d'IntelliNav face aux divergences entre la carte statique et l'environnement réel. Sa capacité à raisonner à l'échelle des catégories d'objets, à évaluer des hypothèses alternatives, à mettre à jour dynamiquement la carte sémantique et à gérer efficacement l'exploration dans des environnements partiellement connus en fait un système adapté à des contextes réels, où les objets peuvent être déplacés ou supprimés sans préavis.

4.4 Conclusion

Ce chapitre a présenté une évaluation des frameworks IntelliMap et IntelliNav, démontrant leur efficacité dans des environnements intérieurs complexes et dynamiques. Les expérimentations ont permis d'analyser les performances de ces deux systèmes à travers plusieurs critères clés, notamment la précision de la cartographie sémantique, la robustesse de la navigation, l'adaptabilité aux changements dynamiques, ainsi que l'efficacité computationnelle.

Dans la première partie, l'évaluation d'IntelliMap a mis en évidence sa capacité à générer des cartes sémantiques précises et hiérarchisées, adaptées à la navigation robotique. Les tests réalisés sur des environnements réels et simulés ont montré que le système peut localiser des objets avec une précision élevée, tout en minimisant l'empreinte mémoire grâce à sa structure optimisée. L'analyse comparative avec d'autres approches a confirmé que IntelliMap offre une couverture sémantique étendue et une efficacité supérieure en matière de stockage.

Dans la seconde partie, l'expérimentation d'IntelliNav a permis d'évaluer la navigation sémantique basée sur ces cartes. Les résultats ont démontré que l'approche proposée permet une planification des trajectoires, en prenant en compte les relations sémantiques entre les pièces et les objets. Grâce à l'intégration des poids dynamiques dans la HSTM, le robot a pu ajuster ses itinéraires en fonction des obstacles rencontrés, garantissant une adaptation continue aux conditions environnementales. L'évaluation de la validation des sous-objectifs a également révélé une réduction significative des distances parcourues, optimisant ainsi l'efficacité de la navigation locale. Enfin, l'analyse des performances computationnelles a montré que le système est capable de générer des trajectoires optimales en quelques millisecondes, ce qui rend possible une navigation fluide en temps réel.

L'ensemble de ces expérimentations confirme la pertinence de l'architecture proposée pour la navigation robotique en milieu dynamique. IntelliMap et IntelliNav offrent une solution complète, intégrant un modèle de cartographie robuste et un système de navigation adaptatif, garantissant une navigation efficace, explicable et résiliente face aux changements environnementaux.

CHAPITRE 5

Conclusion, perspectives et valorisation industrielle

5.1 Synthèse des contributions

Dans ce manuscrit, nous avons proposé IntelliMap et IntelliNav, deux frameworks intégrés visant à améliorer la navigation sémantique des robots mobiles dans des environnements complexes et dynamiques. Nos travaux ont abordé plusieurs défis fondamentaux en cartographie et en navigation, en mettant l'accent sur l'intégration de la sémantique, la mise à jour dynamique des cartes, et l'adaptabilité des chemins de navigation. À travers le développement et l'évaluation des frameworks IntelliMap et IntelliNav, nous avons proposé une solution intégrée qui combine la génération d'une carte sémantique hiérarchique avec un système de navigation adaptatif et explicable. L'objectif principal était de permettre aux robots de naviguer efficacement en exploitant non seulement les informations métriques, mais également les relations sémantiques entre les pièces et les objets, tout en intégrant des mécanismes de mise à jour dynamique et de replanification pour garantir une meilleure adaptation aux changements de l'environnement.

IntelliMap constitue un framework de cartographie hiérarchique modulaire, conçu pour générer automatiquement une représentation unifiée de l'environnement, intégrant de manière structurée des informations métriques et sémantiques. La carte HSTM produit se compose de plusieurs couches interconnectées représentant les objets, les pièces, étages et bâtiments, permettant ainsi de modéliser l'environnement à différents niveaux d'abstraction. Chaque module du pipeline de génération est conçu pour être configurable et indépendant, offrant une flexibilité d'adaptation selon les contraintes de l'environnement cible ou les capacités embarquées du robot. Ce caractère modulaire permet notamment de personnaliser la granularité de la carte HSTM générée en fonction des exigences spécifiques de navigation. Contrairement aux approches traditionnelles basées sur des cartes statiques, IntelliMap ne se contente pas de stocker

des informations statiques sur les objets et les pièces, mais fournit une base de référence exploitable pour la navigation sémantique, facilitant ainsi une compréhension contextuelle accrue.

L'un des piliers de notre contribution est IntelliNav. Toutefois, l'adaptabilité et l'intelligence du système reposent principalement sur IntelliNav, le module responsable de la navigation sémantique et des mises à jour dynamiques de la carte. Contrairement aux approches conventionnelles qui utilisent des cartes fixes, IntelliNav ajuste la HSTM en fonction des obstacles détectés, des objets déplacés et des connexions entre les pièces lors de la navigation. Ce mécanisme de mise à jour dynamique garantit que le robot peut naviguer de manière fiable même dans des environnements où la disposition des objets et des passages change fréquemment. Grâce à son système de pondération adaptative des connexions, IntelliNav favorise les chemins validés, évite les zones incertaines et exclut les passages bloqués. En cas d'échec local, une replanification sémantique vers le même objectif est automatiquement déclenchée. Cette capacité à redéfinir dynamiquement un nouveau chemin évite les échecs prématurés de mission et améliore la résilience du système face à l'imprévu.

L'intégration d'une planification globale sémantique constitue une avancée clé de ce travail. IntelliNav exploite les relations fonctionnelles entre les pièces et les objets pour générer des trajectoires pertinentes et contextuelles. La validation des sous-objectifs garantit que chaque étape du parcours est bien exécutée avant de passer à l'étape suivante, améliorant ainsi la fiabilité et l'explicabilité du processus de navigation. Par ailleurs, nous avons introduit une stratégie d'exploration intelligente dans laquelle le robot estime la localisation probable d'objets non cartographiés en interrogeant un LLM, réduisant ainsi le temps de recherche et optimisant les déplacements.

L'évaluation expérimentale menée dans des environnements variés a permis de démontrer l'efficacité et la pertinence des propositions introduites dans ce manuscrit. L'analyse de la consommation mémoire a révélé qu'IntelliMap permet une réduction significative des besoins en stockage, avec une compression allant jusqu'à 90% par rapport aux approches basées sur des graphes de scène 3D, tout en conservant une couverture sémantique complète. En termes de navigation, IntelliNav a permis une réduction moyenne de 21% à 35% des distances parcourues grâce à son mécanisme de validation des pièces et d'adaptation dynamique des trajectoires.

5.2 Impact industriel et pertinence des choix architecturaux pour l'entreprise euroDAO

Dans le cadre de cette thèse CIFRE réalisée en collaboration avec l'entreprise euroDAO, spécialisée dans la conception et le développement de robots mobiles autonomes adaptés aux exigences particulières des clients, l'un des objectifs majeurs a été de proposer une architecture générale, modulable et adaptable aux contraintes réelles du terrain. Les deux frameworks développés, IntelliMap et IntelliNav, s'inscrivent dans cette logique d'opérationnalisation industrielle, où les exigences peuvent varier d'un client à l'autre, en fonction de la nature des

missions robotiques attendues, de la complexité des lieux et des ressources disponibles.

L'un des apports fondamentaux d'IntelliMap est sa capacité à générer des cartes sémantiques hiérarchiques personnalisables, composées de couches configurables, actuellement pour les objets et pour les pièces et bâtiments. Dans un contexte industriel, où chaque site peut présenter des structures, des contraintes et des objectifs de navigation spécifiques, cette flexibilité est un atout stratégique. Certains ont simplement besoin d'une navigation standard entre pièces, comme pour des robots de surveillance, tandis que d'autres requièrent une détection fine d'objets pour des applications logistiques en entrepôts. D'autres encore ont des besoins complexes de gestion multi-étages, particulièrement dans des environnements comme les hôpitaux ou les centres commerciaux. La carte générée peut s'ajuster au niveau de sémantique requis : une carte orientée uniquement vers les objets pour des missions de collecte, ou bien enrichie des informations topologiques si l'objectif nécessite une navigation structurée entre zones fonctionnelles. Cette capacité à produire des cartes adaptatives offre ainsi à l'entreprise un avantage compétitif : un seul framework peut être déployé sur des sites variés, avec un niveau de personnalisation conforme au cahier des charges de chaque client, sans avoir à redévelopper un pipeline complet pour chaque projet.

Dans un paysage technologique en constante évolution, où de nouveaux modèles de machine learning et de nouveaux algorithmes optimisés émergent régulièrement, il est crucial pour l'entreprise de pouvoir intégrer ces avancées sans devoir restructurer son architecture complète. Notre framework répond à ce besoin grâce à ses modules interchangeables. Le choix méthodologique d'utiliser des modules indépendants aux interfaces bien définies dans les deux frameworks permet une grande liberté d'expérimentation pour l'équipe R&D. Par exemple, dans le module de détection d'objets d'IntelliMap, la méthode de détection utilisée n'est pas figée : tant que la structure de sortie respecte le format attendu par la couche des objets de la HSTM, l'entreprise peut substituer le modèle utilisé [66, 81] sans impacter le reste du système. Ce principe d'abstraction des blocs est également appliqué à la planification de trajets dans IntelliNav. Bien que Dijkstra ait été utilisé pour les expérimentations, rien n'empêche de tester d'autres algorithmes plus performants ou spécifiques (A*, RRT*, navigation LLM-guidée, etc.) selon les besoins du client entre précision, robustesse, vitesse. Ainsi, l'entreprise conserve une liberté méthodologique forte, tout en garantissant l'interopérabilité des composants du système. Cette caractéristique permet à l'entreprise de tester et d'intégrer facilement les dernières avancées technologiques, tout en maintenant la cohérence globale du système. Elle offre également la possibilité de réaliser des benchmarks entre différentes approches algorithmiques pour sélectionner la solution la plus adaptée à chaque cas d'usage client.

La structuration des frameworks en composants interconnectés mais découplés (HSTM, module de navigation, module de mise à jour) permet non seulement une meilleure maintenabilité mais aussi une extensibilité facilitée. De nouveaux blocs fonctionnels peuvent être intégrés ultérieurement (comme des modules de supervision humaine, de collaboration multi-robot, ou des stratégies de mémorisation plus poussées), sans altérer la stabilité de l'architecture existante. Cette ouverture est essentielle dans un contexte industriel où les besoins évoluent rapidement,

et où la capacité à tester rapidement de nouvelles fonctionnalités représente un facteur de différenciation majeur.

Le framework IntelliNav permet d'adapter la navigation du robot à la nature de la mission définie par le client. Qu'il s'agisse d'un transport entre zones fonctionnelles, de la recherche d'un objet spécifique, ou de la validation de points de contrôle, le système peut être configuré pour ajuster : le type de sous-objectifs à suivre (objets, pièces, checkpoints), le niveau de confiance requis pour valider une étape, les règles de replanification et de mise à jour du graphe. De plus, IntelliNav offre une possibilité de personnalisation fine des stratégies de mise à jour : actuellement, les mises à jour du graphe sémantique tiennent compte des connexions bloquées ou débloquées, de la disparition ou du déplacement d'objets, ainsi que de la distinction entre pièces déjà visitées et non visitées. Cette logique peut également être étendue à d'autres types d'événements, en cohérence avec la structure sémantique hiérarchique générée par IntelliMap. Parmi ces extensions possibles figurent le changement d'usage d'une zone, la détection de zones à risque élevé (par exemple, des zones encombrées ou temporairement dangereuses), ou encore la définition de chemins à priorité spécifique pour des cas d'urgence. Des règles temporelles peuvent également être intégrées, comme la restriction d'accès à certaines zones selon l'heure ou le jour (zone accessible uniquement en journée). Ces mécanismes renforcent la flexibilité du système face aux environnements dynamiques et permettent d'adapter la navigation aux contraintes opérationnelles ou contextuelles.

5.2.1 Exemple d'intégration dans un hôpital multi-étages

IntelliMap et IntelliNav pourraient être exploités dans un cas d'usage hospitalier pour répondre à des exigences différenciées selon les étages et les services, illustrant leur adaptabilité potentielle à des environnements complexes et fortement contraints. Dans ce type d'environnement, les attentes varient selon les étages et les services.

Par exemple, à l'étage 1, les missions peuvent consister à accompagner des patients vers des pièces fonctionnelles telles que la salle d'attente, la pneumologie ou le bloc opératoire. Ici, la navigation sémantique repose uniquement sur la couche des pièces : les sous-objectifs sont des salles clairement identifiées, et aucune interaction avec des objets spécifiques n'est nécessaire. La HSTM est donc générée en ne conservant que la couche des pièces et la couche métrique, ce qui réduit la complexité du graphe tout en restant suffisant pour le besoin.

À l'étage 2, en revanche, certaines missions requièrent une navigation plus fine, par exemple pour atteindre un lit désigné dans une chambre, un chariot médical ou un équipement. Dans ce cas, la couche des objets est activée et intégrée à la HSTM, permettant une navigation sémantique ciblée sur des entités spécifiques. IntelliNav adapte alors dynamiquement le type de sous-objectifs à atteindre (pièce ou objet), les règles de replanification en cas d'obstacle (par exemple, accès temporairement bloqué à une pièce ou à un objet) et les critères de validation d'étape (présence dans la pièce ou proximité avec l'objet cible).

Ce cas d'usage démontre la personnalisation fine permise par le framework proposé : le

graphe peut être construit avec un niveau de granularité variable selon les besoins, et les stratégies de navigation sont configurables à la mission. Cette flexibilité est particulièrement pertinente pour l'entreprise, où les environnements opérationnels sont hétérogènes et les exigences métier évolutives.

5.2.2 Exemple d'intégration dans un site industriel

Dans un site industriel combinant zones de production et zones administratives (par exemple, un atelier connecté à un bâtiment administratif), les besoins en navigation robotique varient fortement selon le contexte opérationnel. Un robot mobile peut être amené à effectuer des missions de transport de matériel en zone logistique, de surveillance dans les allées de production, ou encore d'assistance au personnel dans les espaces de bureaux.

Grâce à la flexibilité d'IntelliMap, ces zones aux caractéristiques distinctes peuvent être cartographiées de manière différenciée mais intégrée dans une même Hiérarchical Semantic Topometric Map (HSTM). Par exemple, dans les zones de bureau, la navigation peut se limiter à des objectifs de type "pièces" (ex. : salle de réunion, bureau de direction), sans nécessiter de représentation d'objets individuels. À l'inverse, en zone industrielle, les missions peuvent exiger une granularité plus fine, avec une navigation ciblée vers des objets spécifiques comme des établis, des armoires de sécurité ou des points de contrôle d'équipement. IntelliMap permet alors d'activer la couche des objets uniquement dans les zones où elle est pertinente.

Un point important pour notre partenaire industriel est la capacité à fusionner ces représentations hétérogènes dans une même carte, afin de permettre au robot de naviguer entre ces deux zones. Cette intégration est rendue possible par l'intégration de la représentation de niveau bâtiment (couches des pièces) dans la HSTM, qui permet de structurer logiquement les sous-espaces selon leur fonction (usine, bureaux, entrepôt). IntelliNav utilise cette hiérarchie pour adapter dynamiquement ses stratégies de navigation : choix du niveau sémantique pertinent, ajustement des règles de replanning, et sélection du type de sous-objectif (des pièces, qui correspondant à des zones fonctionnelles ou des objets) selon la nature de la mission.

Cette capacité à composer une représentation modulaire, adaptable et interopérable constitue un levier essentiel pour le déploiement de solutions robotiques dans des contextes industriels complexes et en constante évolution.

Ainsi, les choix effectués dans la conception des frameworks IntelliMap et IntelliNav répondent aux exigences de généricité, d'évolutivité et de personnalisation, essentielles dans un contexte industriel multi-client. L'architecture proposée permet à l'entreprise de proposer une solution adaptable, testable et maintenable, tout en conservant la capacité à intégrer rapidement de nouvelles avancées issues des travaux de recherche ou des retours d'expérience terrain. Ce positionnement hybride entre robustesse industrielle et flexibilité R&D est une condition essentielle pour accompagner durablement les mutations du secteur robotique et pour renforcer la valeur ajoutée des solutions proposées aux clients.

5.3 Perspectives

Les travaux proposés dans ce manuscrit présentent néanmoins certaines limitations qu'il convient de souligner. Tout d'abord, la dépendance à une carte sémantique préexistante peut limiter l'applicabilité du framework dans des environnements entièrement non structurés ou inconnus, où aucune cartographie initiale n'a été réalisée. La gestion des éléments inconnus reste, pour l'instant, limitée à la recherche d'objets non cartographiés dans la carte existante, sans capacité d'adaptation structurelle.

Par ailleurs, la gestion des changements dynamiques, notamment en ce qui concerne les objets très mobiles ou fréquemment déplacés, soulève certaines limites. Par exemple, dans le cas d'objets très instables comme une tasse, l'attribution d'une position fixe dans la carte sémantique devient peu pertinente. Une piste d'amélioration consisterait à différencier les objets en fonction de leur nature statique ou dynamique, en adaptant les stratégies de mise à jour, de cartographie et de navigation en conséquence. Cela permettrait d'optimiser l'effort de mise à jour en privilégiant les éléments pertinents pour la planification.

En outre, les valeurs numériques des poids affectés aux arêtes de la HSTM n'ont pas encore fait l'objet d'une calibration empirique approfondie. Pourtant, ces poids ont une influence significative sur les performances de navigation. Par exemple, dans les scénarios multi-étages, le rapport entre les coûts des connexions inter-étages et la longueur des chemins alternatifs peut déterminer si le robot privilégiera ou non une transition verticale. Il serait donc pertinent de mener une étude plus poussée sur l'influence de ces paramètres et de proposer un processus semi-automatisé de calibration, éventuellement assisté par un expert humain selon les caractéristiques de la plateforme robotique ou du contexte opérationnel (accessibilité des étages, nature des obstacles, etc.).

Un autre axe de développement concerne la gestion de la mémoire historique des missions passées. Comme exposé dans ce manuscrit, IntelliNav enregistre diverses informations utiles à la surveillance à long terme des environnements, telles que les poids des arêtes et des nœuds reflétant les changements récents détectés. Cependant, le processus actuel de gestion de cette mémoire reste simple. Des extensions prometteuses incluraient par exemple des alertes automatiques déclenchées en cas de blocages récurrents sur certaines connexions, pouvant initier soit une mise à jour globale de la carte, soit une requête explicite à un opérateur humain pour inspection ou revalidation. Le framework proposé offre déjà les outils nécessaires pour intégrer de tels mécanismes décisionnels plus avancés, visant à renforcer la fiabilité à long terme dans des environnements dynamiques.

Nous envisageons une validation en conditions réelles, en réalisant des expérimentations pratiques pour évaluer sa robustesse et sa capacité à passer à l'échelle dans des environnements opérationnels variés. Cette évaluation en milieu réel permettra d'analyser l'adaptabilité du cadre face aux défis de la perception robotique et aux conditions bruitées.

Enfin, une perspective spécifique a été explorée dans ce manuscrit avec la proposition de MultiNav, une extension distribuée du framework basée sur la HSTM. Cette architecture théo-

rique multi-robot vise à mutualiser les ressources cartographiques entre plusieurs agents grâce à un mécanisme de synchronisation conditionnelle et à une gestion asynchrone des mises à jour, ouvrant ainsi la voie à une navigation collaborative à grande échelle. Les premières spécifications de cette approche sont présentées en annexe A.

En conclusion, ce travail a introduit une approche intégrée pour la navigation sémantique en environnements intérieurs dynamiques. En combinant une cartographie sémantique hiérarchique (IntelliMap) avec un système de navigation adaptatif et explicable (IntelliNav), nous avons démontré qu'il est possible d'améliorer significativement la capacité des robots à naviguer dans des environnements complexes tout en garantissant une prise de décision robuste et compréhensible. Ces avancées ouvrent la voie à de nombreuses applications, allant de la logistique en milieux industriels à la robotique d'assistance pour les espaces domestiques et les services. Les perspectives futures viseront à renforcer encore davantage l'autonomie et la flexibilité du système, en intégrant des modèles d'apprentissage évolutifs.

Bibliographie

[1] Zhenpeng He, Hao Sun, Jiawei Hou, Yajun Ha, and Sören Schwertfeger. Hierarchical topometric representation of 3d robotic maps. *Auton. Robots*, 45(5) :755–771, 2021.

[2] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11) :915–926, 2008. Semantic Knowledge in Robotics.

[3] Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 116–121. IEEE, 1985.

[4] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3) :249–265, 1987.

[5] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[6] Michael Montemerlo. Fastslam : A factored solution to the simultaneous localization and mapping problem. *Proc. of AAAI02*, 2002.

[7] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial intelligence*, 99(1) :21–71, 1998.

[8] Howie Choset and Joel Burdick. Sensor-based exploration : The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19(2) :96–125, 2000.

[9] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1) :191–233, 2000.

[10] Markus Hiller, Chen Qiu, Florian Particke, Christian Hofmann, and Jörn Thielecke. Learning topometric semantic maps from occupancy grids. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4190–4197, 2019.

[11] Iro Armeni, Zhi-Yang He, Amir Zamir, Junyoung Gwak, Jitendra Malik, Martin Fischer, and Silvio Savarese. 3d scene graph : A structure for unified semantics, 3d space, and camera. pages 5663–5672, Seoul, Korea (South), 2019. IEEE.

[12] Ue-Hwan Kim, Jin-Man Park, Taek jin Song, and Jong-Hwan Kim. 3-D Scene Graph : A Sparse and Semantic Representation of Physical Environments for Intelligent Agents. *IEEE Transactions on Cybernetics*, 50 :4921–4933, 2020.

[13] S. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. SceneGraphFusion : Incremental 3D Scene Graph Prediction From RGB-D Sequences. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7511–7521, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[14] N. Hughes, Y. Chang, and L. Carlone. Hydra : A real-time spatial perception system for 3D scene graph construction and optimization. 2022.

[15] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion : Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021.

[16] Tixiao Shan, Abhinav Rajvanshi, Niluthpol Mithun, and Han-Pang Chiu. Graph2nav : 3d object-relation graph generation to robot navigation. *arXiv preprint arXiv :2504.16782*, 2025.

[17] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3D Dynamic Scene Graphs : Actionable Spatial Perception with Places, Objects, and Humans. In Marc Toussaint, Antonio Bicchi, and Tucker Hermans, editors, *Robotics : Science and Systems XVI*, Oregon, USA, 2020.

[18] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera : From SLAM to spatial perception with 3D dynamic scene graphs. *Int. J. Robotics Res.*, 40(12-14) :1510–1546, 2021.

[19] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics : Hierarchical representations and real-time systems. *The International Journal of Robotics Research*, page 02783649241229725, 2023.

[20] Yun Chang, Luca Ballotta, and Luca Carlone. D-lite : Navigation-oriented compression of 3d scene graphs under communication constraints, 2023.

[21] Nicky Zimmerman, Matteo Sodano, Elias Marks, Jens Behley, and Cyrill Stachniss. Constructing Metric-Semantic Maps Using Floor Plan Priors for Long-Term Indoor Localization. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[22] Guglielmo Gemignani, Roberto Capobianco, Emanuele Bastianelli, Domenico Daniele Bloisi, Luca Iocchi, and Daniele Nardi. Living with robots : Interactive environmental knowledge acquisition. *Robotics and Autonomous Systems*, 78 :1–16, 2016.

[23] Paul Kremer, Hriday Bavle, Jose Luis Sanchez-Lopez, and Holger Voos. S-nav : Semantic-geometric planning for mobile robots. *arXiv preprint arXiv :2307.01613*, 2023.

[24] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33 :4247–4258, 2020.

[25] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE, 2005.

[26] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1) :34–46, 2007.

[27] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros : an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[28] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE international symposium on safety, security, and rescue robotics*, pages 155–160. IEEE, 2011.

[29] Ji Zhang, Sanjiv Singh, et al. Loam : Lidar odometry and mapping in real-time. In *Robotics : Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[30] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of field robotics*, 36(2) :416–446, 2019.

[31] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam : Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.

[32] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2) :1729881419841532, 2019.

[33] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d : A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13154–13164, 2023.

[34] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d : A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[35] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving ? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[36] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes : A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[37] Mohammadreza Sharif, Kiran Mohan, and Sarath Suvarna. 2d floor plan segmentation based on down-sampling. *arXiv preprint arXiv :2303.13798*, 2023.

[38] Serhan Coşar, Manuel Fernandez-Carmona, Roxana Agrigoroaie, Jordi Pages, François Ferland, Feng Zhao, Shigang Yue, Nicola Bellotto, and Adriana Tapus. Enrichme : Perception and interaction of an assistive robot for the elderly at home. *International Journal of Social Robotics*, 12 :779–805, 2020.

[39] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping : Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6) :1309–1332, 2016.

[40] Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr : Building and reasoning over long-horizon spatio-temporal memory for robot navigation. *arXiv preprint arXiv :2409.13682*, 2024.

[41] Quentin Serdel, Julien Marzat, and Julien Moras. Smana : Semantic mapping and navigation architecture for autonomous robots. In Giuseppina Gini, Henk Nijmeijer, and Dimitar P. Filev, editors, *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2023, Rome, Italy, November 13-15, 2023, Volume 1*, pages 453–464. SCITEPRESS, 2023.

[42] Silya Achat, Julien Marzat, and Julien Moras. Path planning incorporating semantic information for autonomous robot navigation. In *19th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2022.

[43] Nico Sun, Erfu Yang, Jonathan Corney, and Yi Chen. Semantic path planning for indoor navigation and household tasks. In *Towards Autonomous Robotic Systems : 20th Annual Conference, TAROS 2019, London, UK, July 3–5, 2019, Proceedings, Part II 20*, pages 191–201. Springer, 2019.

[44] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2) :100–107, 1968.

[45] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra : his life, work, and legacy*, pages 287–290. 2022.

[46] Steven M LaValle and James J Kuffner. Rapidly-exploring random trees : Progress and prospects : Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. *Algorithmic and computational robotics*, pages 303–307, 2001.

[47] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[48] Pedro Núnez, Luis J Manso, Pablo Bustos, Paulo Drews-Jr, and Douglas G Macharet. A proposal for the design of a semantic social path planner using cortex. In *Workshops on Physical Agent*, pages 31–37, 2016.

[49] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-nav : Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. *arXiv preprint arXiv :2409.18794*, 2024.

[50] Yiran Qin, Ao Sun, Yuze Hong, Benyou Wang, and Ruimao Zhang. Navigatediff : Visual predictors are zero-shot navigation assistants. *arXiv preprint arXiv :2502.13894*, 2025.

[51] Yao Cheng, Fengyang Jiang, Zhe Han, Huaizhen Wang, Fengyu Zhou, Zhaochuan Li, Bin Wang, and Yang Huang. Robot navigation based on 3d scene graphs with the llm tooling*. In *2024 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 476–482, 2024.

[52] Daniel Honerkamp, Martin Büchner, Fabien Despinoy, Tim Welschehold, and Abhinav Valada. Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation, July 2024. arXiv :2403.08605.

[53] Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Sünderhauf, Feras Dayoub, and Ian Reid. RoboHop : Segment-based Topological Map Representation for Open-World Visual Navigation, May 2024. arXiv :2405.05792.

[54] Zhirui Dai, Arash Asgharivaskasi, Thai Duong, Shusen Lin, Maria-Elizabeth Tzes, George Pappas, and Nikolay Atanasov. Optimal scene graph planning with large language model guidance. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14062–14069, 2024.

[55] Dhruv Shah, Michael Equi, Blazej Osinski, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with Large Language Models : Semantic Guesswork as a Heuristic for Planning, October 2023. arXiv :2310.10103 [cs].

[56] Rui Fukushima, Kei Ota, Asako Kanezaki, Yoko Sasaki, and Yusuke Yoshiyasu. Object memory transformer for object goal navigation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11288–11294. IEEE, 2022.

[57] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson : Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35 :32340–32352, 2022.

[58] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM : Vision-Language Frontier Maps for Zero-Shot Semantic Navigation, December 2023. arXiv :2312.03275.

[59] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. SOON : Scenario Oriented Object Navigation with Graph-based Exploration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12684–12694, Nashville, TN, USA, June 2021. IEEE.

[60] Zhe Ni, Xiaoxin Deng, Cong Tai, Xinyue Zhu, Qinghongbing Xie, Weihang Huang, Xiang Wu, and Long Zeng. GRID : Scene-Graph-based Instruction-driven Robotic Task Planning, March 2024. arXiv :2309.07726.

[61] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan : Grounding large language models using 3d scene graphs for scalable task planning. In *7th Annual Conference on Robot Learning*, 2023.

[62] Manjunath Narayana, Andreas Kolling, Lucio Nardelli, and Phil Fong. Lifelong update of semantic maps in dynamic environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6164–6171. IEEE, 2020.

[63] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object goal navigation with recursive implicit maps. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[64] Maxim Likhachev and Sven Koenig. Lifelong planning for mobile robots. In *Advances in Plan-Based Control of Robotic Agents : International Seminar Dagstuhl Castle, Germany, October 21–26, 2001 Revised Papers*, pages 140–156. Springer, 2003.

[65] Rey Reza Wiyatno, Anqi Xu, and Liam Paull. Lifelong topological visual navigation. *IEEE Robotics and Automation Letters*, 7(4) :9271–9278, 2022.

[66] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*, 2020.

[67] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster r-cnn. *https ://github.com/jwyang/faster-rcnn.pytorch*, 2017.

[68] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN : Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[69] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[70] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[71] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini : a family of highly capable multimodal models. *arXiv preprint arXiv :2312.11805*, 2023.

[72] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv :2401.04088*, 2024.

[73] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1 : Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv :2501.12948*, 2025.

[74] Moveh Samuel, Mohamed Hussein, and Maziah Binti Mohamad. A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *International Journal of Computer Applications*, 135(1) :35–38, 2016.

[75] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control : Theory and practice—a survey. *Automatica*, 25(3) :335–348, 1989.

[76] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d) : 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[77] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[78] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs : Open-vocabulary 3d scene graphs for perception and planning. *arXiv*, 2023.

[79] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In *Proceedings of Robotics : Science and Systems*, Delft, Netherlands, July 2024.

[80] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[81] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[82] Siyan Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. Multi-robot collaborative dense scene reconstruction. *ACM Transactions on Graphics (TOG)*, 38(4) :1–16, 2019.

[83] Erkan Uslu, Furkan Cakmak, Nihal Altuntaş, Salih Marangoz, Mehmet Fatih Amasyalı, and Sırma Yavuz. An architecture for multi-robot localization and mapping in the gazebo/robot operating system simulation environment. *Simulation*, 93(9) :771–780, 2017.

[84] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi : Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 38(4), 2022.

[85] Xu Liu, Jiuzhou Lei, Ankit Prabhu, Yuezhan Tao, Igor Spasojevic, Pratik Chaudhari, Nikolay Atanasov, and Vijay Kumar. Slideslam : Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation. *arXiv preprint arXiv :2406.17249*, 2024.

[86] Yufeng Yue, Mingxing Wen, Chunyang Zhao, Yuanzhe Wang, and Danwei Wang. Cosem : Collaborative semantic map matching framework for autonomous robots. *IEEE Transactions on Industrial Electronics*, 69(4) :3843–3853, 2021.

[87] Miguel Fernandez-Cortizas, Hriday Bavle, David Perez-Saura, Jose Luis Sanchez-Lopez, Pascual Campoy, and Holger Voos. Multi s-graphs : an efficient distributed semantic-relational collaborative slam. *IEEE Robotics and Automation Letters*, 2024.

[88] Yufeng Yue, Chunyang Zhao, Ruilin Li, Chule Yang, Jun Zhang, Mingxing Wen, Yuanzhe Wang, and Danwei Wang. A hierarchical framework for collaborative probabilistic semantic

mapping. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 9659–9665. IEEE, 2020.

[89] Xianzhe Zhao, Shiliang Shao, Ting Wang, Chuxi Fang, Jin Zhang, and Hai Zhao. A review of multi-robot collaborative simultaneous localization and mapping. In *2023 IEEE International Conference on Unmanned Systems (ICUS)*, pages 900–905. IEEE, 2023.

[90] Abdessalem Achour, Hiba Al-Assaad, Yohan Dupuis, and Madeleine El Zaher. Collaborative mobile robotics for semantic mapping : A survey. *Applied Sciences*, 12(20) :10316, 2022.

[91] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 476–481. IEEE, 2000.

[92] Yuming Ning, Tuanjie Li, Cong Yao, Wenqian Du, and Yan Zhang. Hms-rrt : A novel hybrid multi-strategy rapidly-exploring random tree algorithm for multi-robot collaborative exploration in unknown environments. *Expert Systems With Applications*, 247 :123238, 2024.

[93] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-navgpt : Multi-robot cooperative visual semantic navigation using large language models. *arXiv preprint arXiv :2310.07937*, 2023.

[94] Samarth Kalluraya, George J Pappas, and Yiannis Kantaros. Multi-robot mission planning in dynamic semantic environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1630–1637. IEEE, 2023.

[95] Yiannis Kantaros and George J Pappas. Optimal temporal logic planning for multi-robot systems in uncertain semantic maps. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4127–4132. IEEE, 2019.

[96] Javier Yu, Joseph A Vincent, and Mac Schwager. Dinno : Distributed neural network optimization for multi-robot collaborative learning. *IEEE Robotics and Automation Letters*, 7(2) :1896–1903, 2022.

ANNEXE A

MultiNav, navigation multi-robot distribuée basée sur la HSTM

Cette section présente MultiNav, une perspective avancée étudiée de manière théorique, qui propose une extension distribuée du système de navigation sémantique basé sur la HSTM à un ensemble de robots collaboratifs.

A.1 Introduction

Dans les environnements dynamiques de grande envergure, tels que les entrepôts, les hôpitaux ou les bâtiments multi-étages, une seule unité robotique peut s'avérer insuffisante pour garantir une couverture spatiale rapide et efficace des missions à effectuer. L'utilisation simultanée de plusieurs robots permet une répartition stratégique des tâches, soit en affectant chaque robot à une zone ou à un étage spécifique, soit en autorisant plusieurs robots à parcourir simultanément le même espace pour améliorer la précision des observations, résoudre des ambiguïtés ou accélérer l'exécution de certaines tâches. Cette organisation multi-robot facilite une gestion stratégique des missions, tout en s'adaptant efficacement aux contraintes des grands espaces dynamiques. Toutefois, elle nécessite la mise en place d'un système distribué de partage des connaissances et de mise à jour cartographique, capable de garantir la cohérence globale des informations et une gestion efficace des éventuels conflits entre robots. Dans des environnements intérieurs dynamiques de grande échelle, l'utilisation de plusieurs robots coopérant accroît l'efficacité opérationnelle, permet d'améliorer la couverture spatiale et d'assurer une certaine redondance dans les missions critiques. Cette redondance fonctionnelle se traduit par la capacité du système à tolérer les défaillances partielles ou les erreurs locales, en s'appuyant sur des observations croisées entre robots.

Toutefois, cette configuration multi-robot soulève des défis majeurs en matière de partage d'information, de synchronisation cartographique et de résolution des conflits d'observations

pouvant émerger de perceptions contradictoires ou d'incertitudes liées à l'environnement.

Afin de répondre à ces contraintes, nous proposons une extension multi-agent de notre système de navigation sémantique, fondée sur une architecture hybride distribuée articulée autour d'une carte hiérarchique partagée, la HSTM. Chaque robot dispose d'une copie locale de cette carte, qu'il met à jour indépendamment en fonction de ses propres observations. Les mises à jour ne sont synchronisées avec la carte globale que lorsque certaines conditions spécifiques sont satisfaites, ce qui autorise une gestion distribuée et asynchrone de l'information, tout en réduisant les coûts de communication. Cette approche combine trois objectifs clés. D'abord, l'autonomie locale, où chaque robot fonctionne de manière indépendante selon ses propres perceptions. Ensuite, la cohérence globale, obtenue par l'alignement progressif des cartes locales à l'aide de mécanismes de synchronisation conditionnelle. Enfin, la robustesse collective, qui repose sur des stratégies de validation croisée, de résolution des conflits entre robots, et sur la possibilité de solliciter un opérateur humain en cas d'ambiguïtés persistantes. Ce chapitre présente l'architecture distribuée proposée, ses modules fonctionnels internes, les mécanismes de synchronisation, ainsi que les stratégies de gestion des conflits dans un cadre multi-robot.

A.2 Etat de l'art

L'utilisation de flottes de robots pour l'exploration, la cartographie et la navigation autonome suscitent un intérêt croissant, notamment dans des contextes où la coordination multi-agent permet d'augmenter la couverture spatiale, la robustesse des estimations et la résilience face aux défaillances individuelles. Cette tendance s'appuie sur les avancées récentes en SLAM, en planification collaborative et en intégration de connaissances sémantiques, notamment grâce à l'usage émergent de LLM.

Les approches existantes peuvent être classées en deux grandes catégories. D'une part, les méthodes traditionnelles basées sur des représentations purement géométriques, qui utilisent des cartes métriques comme les grilles d'occupation ou les nuages de points, combinées à des algorithmes de coordination fondés sur des règles explicites pour l'évitement d'obstacles ou le partitionnement de l'espace. D'autre part, les approches plus récentes intégrant des couches sémantiques, qui enrichissent ces représentations avec des informations contextuelles sur les objets et leurs relations, permettant ainsi une prise de décision plus intuitive et collaborative.

Cette section est articulée selon trois axes fonctionnels majeurs : la reconstruction ou cartographie collaborative, l'exploration coordonnée et la navigation sémantique distribuée. Certains travaux relèvent également d'une catégorie transversale, axée sur l'optimisation distribuée ou la planification logique.

A.2.1 Reconstruction dense et cartographie collaborative

Plusieurs approches se concentrent sur la construction collaborative d'une carte géométrique ou sémantique, soit pour la visualisation, soit pour la planification. On distingue notamment

les travaux orientés reconstruction dense, comme [82], qui exploitent une assignation de tâches optimisée par transport de masse pour permettre à plusieurs robots de reconstruire une scène intérieure de manière dense et rapide. D'autres travaux, comme [83], proposent des architectures de SLAM multi-robot reposant sur des implémentations classiques (gMapping, Hector SLAM) dans ROS/Gazebo, mettant en avant les bénéfices de la coopération pour la localisation en milieu structuré.

En complément, plusieurs contributions récentes cherchent à étendre le SLAM classique en y intégrant des informations sémantiques de manière distribuée. Parmi elles, Kimera-Multi [84] et SlideSLAM [85] illustrent des approches robustes et temps réel pour des robots terrestres et aériens, exploitant la détection d'objets et la reconnaissance sémantique pour améliorer la consistance globale. D'autres méthodes, comme COSEM [86] ou Multi S-Graphs [87], utilisent des graphes sémantiques pour aligner ou fusionner des cartes locales distribuées. Le travail [88] propose un cadre hiérarchique probabiliste de cartographie sémantique, structurant l'information à plusieurs niveaux pour faciliter la fusion collaborative.

Deux contributions se démarquent par leur nature de revue de littérature : [89] propose une synthèse des approches collaborative SLAM multi-robot. Ce travail offre une perspective complète sur l'état actuel du domaine, identifiant trois défis majeurs : la synchronisation des estimations locales, l'optimisation des échanges de données dans des conditions de bande passante limitée, et la gestion des incertitudes corrélées. De son côté, [90] offre une vue d'ensemble sur la cartographie sémantique mobile en contexte collaboratif, en discutant les architectures, les modalités de fusion, et les limites actuelles.

A.2.2 Exploration multi-robot d'environnements inconnus

L'exploration multi-robot constitue un problème fondamental en robotique mobile, avec pour objectif principal de maximiser la couverture spatiale tout en minimisant les redondances et les temps de mission. Les travaux pionniers dans ce domaine, tels que ceux de [91], ont établi les bases des stratégies décentralisées basées sur le concept d'utilité des zones inconnues. Ces approches assignent dynamiquement aux robots les régions frontières les plus prometteuses en termes d'information gain. Elles favorisent une exploration efficace sans nécessiter de coordination centralisée.

Les méthodes contemporaines ont considérablement enrichi ce cadre initial. Par exemple, l'algorithme HMS-RRT [92] représente une avancée significative en combinant plusieurs stratégies complémentaires : une version adaptative de RRT (Rapidly-exploring Random Tree) pour la génération de trajectoires, des diagrammes de Voronoï pour le partitionnement dynamique de l'environnement et un mécanisme de marché pour l'allocation optimale des tâches. Cette hybridation permet d'atteindre des performances remarquables.

Cependant, ces méthodes montrent leurs limites face à des dynamiques environnementales imprévisibles. La présence d'objets mobiles ou d'agents non coopératifs peut en effet compromettre l'efficacité des stratégies purement géométriques, soulignant la nécessité d'intégrer des

capacités de perception et de raisonnement plus sophistiquées.

A.2.3 Navigation et planification multi-robot avec raisonnement sémantique

Les travaux les plus récents déplacent l'attention vers des objectifs de plus haut niveau, tels que la navigation collaborative dans des environnements sémantiques ou dynamiques. Des approches comme Multi S-Graphs [87], Kimera-Multi [84] ou SlideSLAM [85] combinent SLAM sémantique et collaboration distribuée. Le cadre Co-NavGPT [93] introduit une innovation majeure en exploitant des LLMs pour générer des plans de navigation visuelle collaborative à partir de prompts structurés, montrant l'émergence de la synergie LLM/robotique multi-agent.

D'autres travaux s'intéressent à la planification logique sémantique collaborative, à travers des spécifications exprimées en logique temporelle linéaire (LTL). C'est le cas de [94] ou [95], qui proposent des méthodes robustes pour planifier des missions complexes sous incertitude, en exploitant les cartes sémantiques construites par les agents.

Enfin, une direction émergente est représentée par DiNNO [96], qui propose un cadre d'apprentissage collaboratif distribué (sans partage de données brutes) basé sur la synchronisation des poids de réseaux de neurones entre robots. Bien que ne se focalisant pas directement sur la navigation ou la cartographie, ce type d'approche pourrait être intégré à des systèmes multi-robots apprenants à long terme.

A.2.4 Limite des approches et proposition

Contrairement aux approches existantes focalisées sur la cartographie ou la coordination par fusion centralisée, notre proposition vise à structurer une navigation multi-robot sémantique distribuée fondée sur une carte hiérarchique (HSTM) partagée. Chaque robot dispose d'une carte locale indépendante, tandis qu'une synchronisation conditionnelle diffuse ou intègre les modifications jugées significatives. L'architecture intègre en outre un module de résolution des conflits (vote, consensus, intervention humaine), ainsi qu'un mécanisme de visualisation collective des états à l'échelle du système.

Cette approche comble un vide dans l'état de l'art en proposant une structure explicite, distribuée et hiérarchique, adaptée à la navigation collaborative dans des environnements intérieurs dynamiques, là où les représentations 3D denses ou les graphes sémantiques globaux montrent des limites en termes de charge computationnelle et de gestion d'incertitude.

A.3 Architecture

L'architecture proposée repose sur une organisation distribuée dans laquelle plusieurs robots coopérant partagent une connaissance hiérarchique et sémantique de l'environnement, représentée par la carte HSTM. Cette carte, commune à l'ensemble du système, est structurée en

une version globale, accessible à tous les robots, et des copies locales, maintenues individuellement par chaque robot et adaptées en fonction de leurs perceptions respectives. Comme illustré dans la figure 32, chaque robot Ri possède sa propre instance de la carte, notée HSTM_Ri, qu'il utilise pour naviguer de manière autonome grâce à son module de navigation sémantique IntelliNav. Ce module exploite la structure hiérarchique de la carte pour planifier des trajectoires sémantiques vers des objectifs définis à différents niveaux (pièces, objets), et est capable d'effectuer des ajustements locaux en fonction des obstacles ou changements détectés. Afin d'assurer une cohérence entre les différentes vues locales tout en minimisant les échanges de données inutiles, chaque robot est équipé d'un module interne appelé Sync Decision. Ce module surveille en continu l'état de la carte locale et détecte les événements significatifs susceptibles de nécessiter une mise à jour de la carte globale. Lorsqu'une condition de synchronisation est remplie, une procédure de synchronisation conditionnelle est déclenchée. Celle-ci distingue deux opérations :

— L'envoi des données, qui correspond à la transmission des mises à jour locales pertinentes vers la carte HSTM globale ;
— La récupération des données, qui consiste à intégrer les mises à jour globales validées par d'autres robots dans la carte locale, garantissant ainsi un alignement progressif des connaissances.

Cette architecture inclut également un mécanisme de gestion des conflits, essentiel dans un contexte multi-robot où les perceptions peuvent diverger. Lorsqu'un désaccord est détecté entre plusieurs robots par exemple, si l'un considère une connexion comme bloquée tandis qu'un autre la juge praticable un processus de résolution est activé. Celui-ci repose d'abord sur une stratégie de consensus vote majoritaire. Si le conflit persiste ou si l'incertitude est trop élevée, un module de requête humaine (HRI : Interaction Humain-Robot) est sollicité afin de faire intervenir un opérateur. Cette intervention humaine garantit une prise de décision fiable dans les cas critiques, tout en maintenant la sécurité et la robustesse du système.

Cette architecture assure un équilibre entre l'autonomie locale des robots, leur capacité à partager et fusionner les informations de manière conditionnelle, et la prise en compte de l'incertitude via une interaction humain-robot en cas de conflit non résolu automatiquement. Elle constitue une base solide pour le déploiement de flottes robotisées dans des environnements intérieurs dynamiques, hétérogènes et partiellement inconnus.

Toutefois, un suivi rigoureux des modifications, une résolution efficace des conflits et une traçabilité fiable des informations échangées dans cette architecture distribuée exigent quelques ajustements spécifiques de la structure initiale de la HSTM.

A.3.1 Adaptation de la structure de la HSTM au contexte multi-robot

Pour supporter efficacement le système multi-robot distribué décrit précédemment, plusieurs ajustements spécifiques ont été intégrés à la structure de la carte HSTM. Ces modifications sont

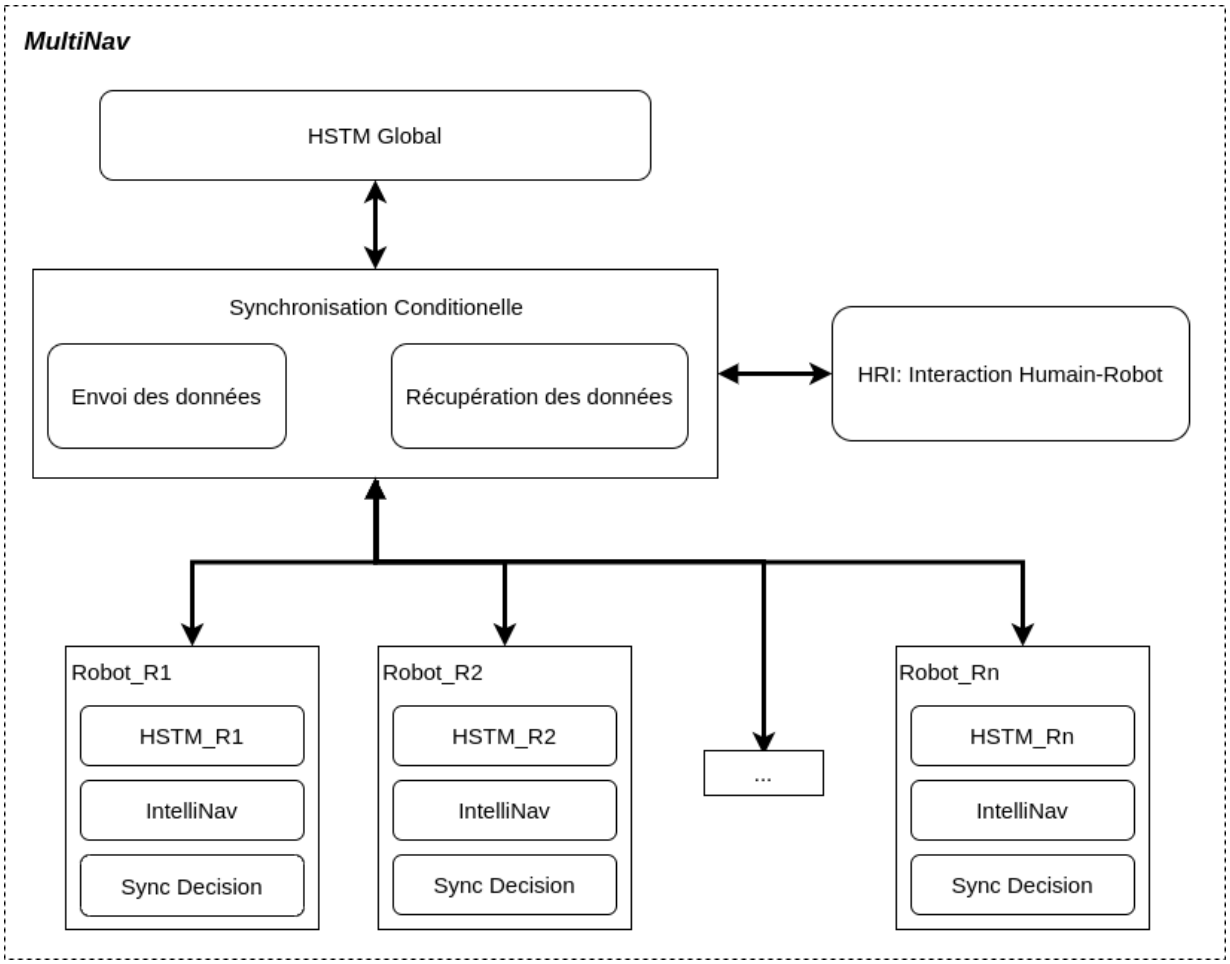


FIGURE 32 – Architecture distribuée proposée pour la navigation multi-robot basée sur la HSTM.

nécessaires pour assurer une gestion rigoureuse des mises à jour distribuées, la résolution fiable des conflits entre robots, et la traçabilité des informations échangées. Plus précisément, la structure initiale de la HSTM a été enrichie par les attributs suivants :

- Identifiant de source (robot ID) : Chaque modification enregistrée dans la carte globale ou locale est associée à un identifiant unique représentant le robot à l'origine de l'observation. Cet attribut assure une traçabilité précise des données et facilite l'analyse de leur provenance en cas de conflit ou d'ambiguïté. La responsabilité des observations est explicitement attribuée à chaque robot, facilitant ainsi l'analyse post-mission et assurant une meilleure transparence.
- Timestamp de modification : Chaque mise à jour effectuée sur la HSTM globale par un robot est accompagnée d'un horodatage indiquant le moment exact de sa détection. Cet attribut temporel est fondamental pour la gestion des conflits et la synchronisation entre cartes locales et globales. En cas de contradiction entre plusieurs versions d'une même entité, le timestamp indique la mise à jour la plus récente. Le système peut ainsi résoudre efficacement les divergences chronologiques des données.
- Statut de validation : Un nouvel attribut indiquant le statut courant de chaque élément de la carte a été ajouté. Ce statut peut prendre plusieurs valeurs :
  - Local : la modification a été effectuée par un robot et n'a pas encore été transmise ou validée au niveau global.
  - Validée globalement : la mise à jour a été acceptée par la carte globale, soit par consensus majoritaire, soit après validation humaine.
  - Conflit : l'élément présente des contradictions entre différentes cartes locales, nécessitant une résolution explicite.
- Mécanisme de version différenciée : Un mécanisme de gestion de versions différenciées est intégré à la HSTM afin de comparer efficacement les écarts entre les versions locales des robots et la carte globale. Ainsi, lors des opérations de synchronisation, seules les différences significatives sont échangées, optimisant ainsi le volume de communication nécessaire. Ce mécanisme facilite également le retour en arrière en cas d'erreurs ou d'incidents, en permettant une restauration rapide vers des états antérieurs clairement identifiés.

Ces attributs additionnels à la structure initiale de la HSTM assurent une traçabilité rigoureuse des données, une résolution efficace des conflits et une gestion optimale des échanges entre les robots. L'intégration de ces modifications est essentielle pour garantir la transparence du système multi-robot proposé.

A.3.2 Carte HSTM locale

Chaque robot maintient une copie locale de la carte hiérarchique HSTM, notée HSTM_Ri. Cette carte représente la structure de l'environnement à plusieurs niveaux sémantiques : des pièces (niveau topologique), aux objets (niveau sémantique), jusqu'aux données locales liées

à la navigation (niveau métrique ou de coût). La HSTM locale est continuellement mise à jour en fonction des observations réalisées par le robot. Elle s'adapte de manière autonome à l'environnement immédiat, même en l'absence de connexion constante avec les autres robots ou la carte globale. La carte locale agit comme une mémoire épisodique, propre à chaque agent. Elle conserve non seulement la structure spatiale observée, mais également les états associés aux éléments (ex. : un chemin est bloqué, un objet est absent), ainsi que des métadonnées telles que l'identifiant du robot, l'horodatage de l'observation. Cette organisation formalise le stockage local des expériences, au service à la fois de la navigation et du processus de synchronisation conditionnelle. La HSTM locale sert donc de support fondamental à toutes les prises de décision internes du robot.

A.3.3 Module de navigation IntelliNav

Le module IntelliNav est responsable de la navigation sémantique autonome du robot à partir de la carte HSTM locale. Il génère un plan global sous forme de chemin sémantique, c'est-à-dire une suite de pièces ou d'objets à traverser ou atteindre, en exploitant la hiérarchie de la carte. Ce plan est ensuite divisé en sous-objectifs locaux (subgoals), facilitant une exécution progressive et adaptative. Lors de l'exécution, le robot suit ce chemin en validant successivement chaque sous-objectif, et déclenche une replanification si des obstacles imprévus ou des changements d'état sont rencontrés. IntelliNav inclut un mécanisme de mémorisation des difficultés rencontrées, comme des chemins temporairement bloqués, des objets déplacés ou des incohérences entre la carte et l'environnement réel. Cette mémoire influence les choix futurs de planification, en modifiant dynamiquement les coûts associés aux trajets. Grâce à cette capacité d'adaptation, le robot peut éviter les zones problématiques et privilégier des alternatives plus robustes. En s'appuyant sur la HSTM locale enrichie, IntelliNav assure ainsi une navigation efficace et résiliente.

A.3.4 Module de décision de synchronisation locale (Sync Decision)

Au niveau local, le module Sync Decision surveille en continu les modifications apportées à la carte HSTM locale, suite aux observations directes du robot dans son environnement. Il a pour rôle de décider de manière autonome si certaines de ces modifications justifient l'envoi d'une demande de synchronisation vers la carte HSTM globale, gérée par le module Synchronisation conditionnelle.

Afin de limiter les communications inutiles et de ne remonter que les changements à fort impact sur la planification globale, la décision de synchronisation repose sur un ensemble restreint de règles. Plus précisément, seules les transitions d'état jugées critiques c'est-à-dire celles pouvant influencer significativement la structure navigable ou éviter des blocages futurs donnent lieu à une demande d'actualisation globale.

Les transitions suivantes sont considérées comme suffisamment importantes pour déclencher une synchronisation par rapport à cette liste de transition dans IntelliNav présenté sur le

tableau A.1 :

Transition	**Synchronisation globale ?**
État 0 → État 3 : (Non testé → Bloqué)	**Oui**
État 1 → État 3 : (Validé → Bloqué)	**Oui**
État 2 → État 1 : (Potentiellement bloqué → Validé)	**Oui**
État 3 → État 2 : (Bloqué → Potentiellement bloqué)	Non
État 2 → État 3 : (Potentiellement bloqué → Bloqué)	Non

TABLE A.1 – Transitions locales déclenchant une synchronisation vers la HSTM globale

— État 0 → État 3 (Non testé → Bloqué) : détection d'un obstacle permanent sur une connexion inconnue. Cette information est essentielle pour éviter que d'autres robots ne planifient des trajets passant par ce chemin.
— État 1 → État 3 (Validé → Bloqué) : requalification d'un chemin précédemment considéré comme sûr vers un état bloqué. Ce changement critique doit être communiqué rapidement afin d'éviter des tentatives d'utilisation par d'autres robots.
— État 2 → État 1 (Potentiellement bloqué → Validé) : un chemin précédemment douteux est confirmé comme praticable. Cette mise à jour améliore la connectivité globale et peut influencer positivement les futurs calculs de trajectoire.

Les autres transitions, telles que Bloqué → Potentiellement bloqué ou Potentiellement bloqué → Bloqué, sont considérées comme transitoires ou locales. Elles sont traitées en interne et mises à jour en fin de mission, sans être immédiatement remontées au niveau global, sauf en cas de confirmation définitive via des observations croisées.

Lorsqu'une condition de synchronisation est remplie, le module Sync Decision génère une requête structurée incluant l'identifiant du robot, la nature du changement et un horodatage. Cette requête est transmise au module Synchronisation conditionelle, qui analysera sa pertinence, vérifiera d'éventuels conflits, et actualisera la carte globale si nécessaire.

En retour, dès qu'une mise à jour est validée au niveau global, chaque robot reçoit une version différente de la HSTM intégrant uniquement les changements confirmés. Le module local applique alors ces différences à la carte locale de manière incrémentale, assurant ainsi une cohérence progressive du système distribué sans nécessiter de synchronisation complète.

A.3.5 Module de Synchronisation conditionnelle

Le module de synchronisation conditionnelle joue un rôle central dans la coordination entre les représentations locales détenues par chaque robot et la version globale de la carte HSTM. Son objectif est d'assurer une cohérence collective, sans compromettre l'autonomie locale des agents ni surcharger le réseau de communication avec des mises à jour redondantes ou contradictoires. Placé entre la carte globale et les robots, ce module est responsable de l'analyse, de la validation et de la diffusion des mises à jour transmises par les différents agents.

Chaque demande de mise à jour émise par un robot contient un ensemble de métadonnées structurées : l'identifiant du robot émetteur, le type exact de transition observée (par exemple, de « validé » à « bloqué »), ainsi qu'un horodatage précis de l'événement. À réception d'une demande, le module SYNC entame une procédure de validation basée sur deux principes fondamentaux :

— **La vérification de cohérence temporelle** : elle consiste à examiner les observations contradictoires reçues pour une même arête dans une fenêtre temporelle glissante de durée τ exprimée en secondes.

— **La règle de majorité** : en cas de divergence, l'état le plus fréquemment rapporté est privilégié, sauf si les observations sont parfaitement équilibrées.

A.3.5.1 Traitement des transitions et envoi vers la carte HSTM globale :

Les transitions suivantes sont traitées de manière conditionnelle et peuvent entraîner une mise à jour de la carte HSTM globale, selon les règles ci-dessous :

— **Transition vers l'état bloqué (non testé ou validé → bloqué)** : Lorsque cette transition est reçue, le module SYNC vérifie qu'aucun autre robot n'a rapporté un état contradictoire (vers « validé ») pour le même chemin dans la fenêtre temporelle τ. Si aucune contradiction n'est détectée, l'état du chemin est requalifié comme potentiellement bloqué sur la carte globale. Cette requalification conserve une mémoire du blocage tout en laissant la possibilité d'une réutilisation ultérieure.

— **Transition vers l'état validé (potentiellement bloqué → validé)** : Si l'observation reçue n'est pas contredite par une transition inverse (vers « bloqué ») au sein de l'intervalle τ , le chemin est mis à jour en « validé » sur la HSTM globale.

— **Gestion des conflits** : Si deux transitions contradictoires sont reçues pour une même arête dans la fenêtre τ, le système entre en état de conflit :

  — Si une majorité claire se dégage parmi les robots ayant observé ce chemin, l'état majoritaire est retenu. Si une majorité relative significative soutient une même version de l'information, celle-ci est retenue et appliquée à la carte globale. En s'appuyant sur l'accord collectif, cette politique tolère certaines erreurs individuelles sans compromettre la cohérence globale.

  — Si les votes sont équilibrés (ex. : 50/50), la zone concernée est marquée avec un attribut de conflit, et une vérification est déléguée à un opérateur humain via le module HRI. Un indicateur d'ambiguïté est ajouté dans la carte HSTM globale, ce qui signale à tous les robots que la zone ou l'entité concernée est sujette à controverse. Ce mécanisme préserve la transparence des données tout en évitant des décisions hâtives ou erronées.

Toutes les décisions prises par cette module sont accompagnées d'un enregistrement horodaté et attribué, assurant une traçabilité complète des mises à jour.

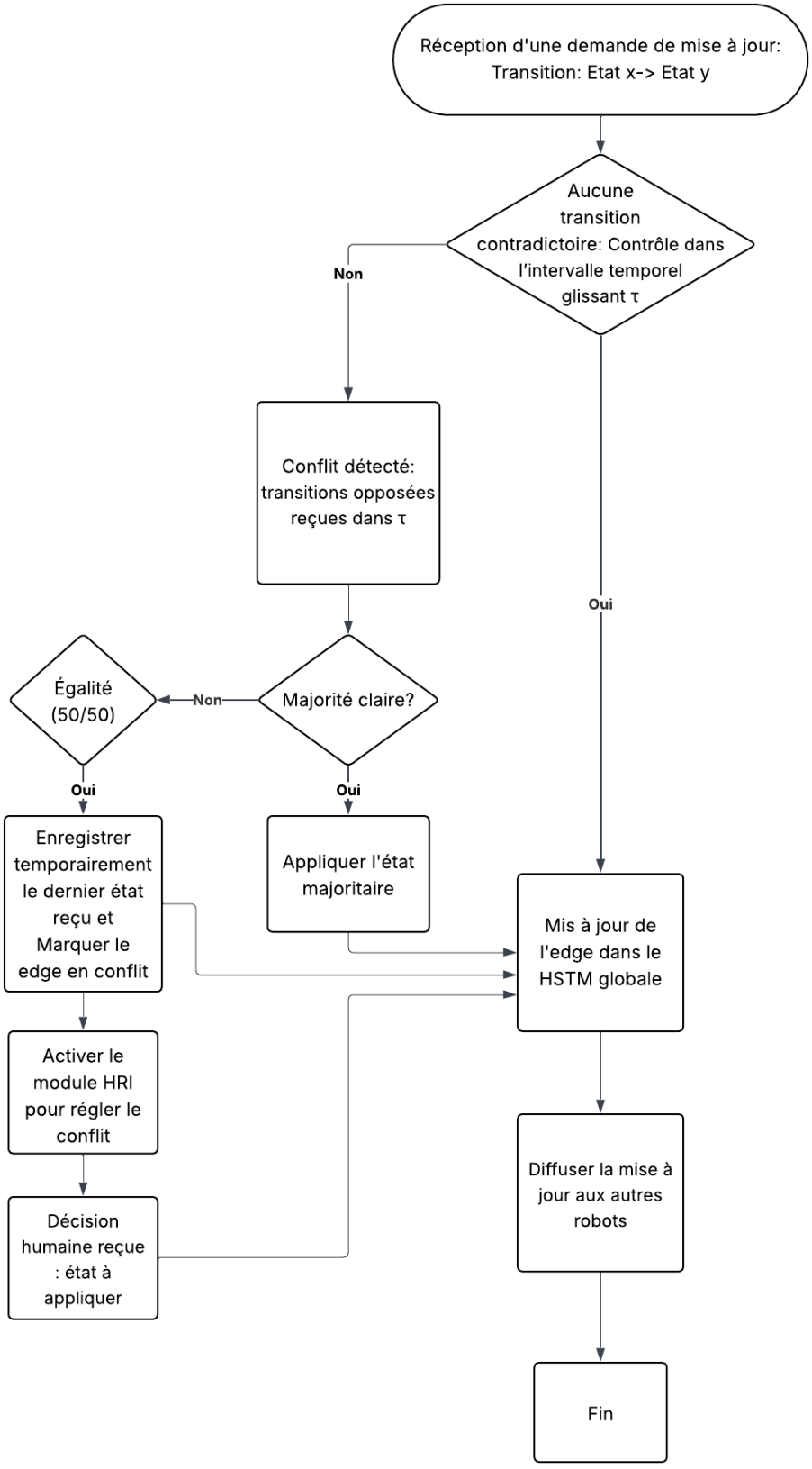


FIGURE 33 – Processus de décision du module de synchronisation pour la mise à jour conditionnelle de la HSTM globale

A.3.5.2 Téléchargement des mises à jour vers les cartes locales :

Une fois qu'un changement est validé, celui-ci est propagé aux autres robots sous forme de mise à jour différentielle, limitant les transferts aux seules portions de la carte ayant été effectivement modifiées. Un attribut de version est attribué à l'HSTM globale. Chaque robot est également responsable d'intégrer dans sa carte locale les modifications validées au niveau global. Ce processus de téléchargement est effectué de manière incrémentale : seules les différences entre la carte locale et la carte globale sont transférées et fusionnées, selon un mécanisme de mise à jour différentielle. Lorsqu'une mise à jour validée est disponible, elle est transmise à l'ensemble des robots. Le module Sync decision de chaque robot réceptionne ces données, les compare à la version locale, puis applique les mises à jour s'il n'y a pas de conflit. Ce mécanisme garantit la cohérence progressive des cartes sans nécessiter de synchronisation complète ni de verrouillage centralisé.

A.3.5.3 Exemple concret de mise à jour et gestion temporelle des conflits :

Prenons l'exemple d'un robot R1 explorant un couloir dans lequel un meuble a été récemment déplacé, obstruant temporairement une connexion entre deux pièces adjacentes. En analysant sa carte HSTM locale, le robot détecte que cette connexion, précédemment marquée comme validée, est désormais infranchissable.

Le module Sync Decision de R1 identifie cette transition comme critique (validé→ bloqué) et génère une demande de synchronisation. Cette demande est transmise au module SYNC, accompagnée de métadonnées comprenant :

— l'identifiant de l'émetteur (ici, R1) ;

— l'horodatage de la détection ;

— la nature exacte du changement observé.

À réception, le module Synchronisation conditionnelle applique les règles de validation définies. Il consulte l'historique des mises à jour reçues au cours des dernières τ=60 secondes pour cette connexion. Si aucune transition contradictoire (vers « validé ») n'a été signalée dans cet intervalle, le changement est accepté, et l'état du chemin est requalifié comme potentiellement bloqué sur la carte HSTM globale. Cette information devient alors accessible à tous les robots du système.

Peu de temps après, un autre robot, R2, s'engage sur le même couloir et, à son tour, inspecte la même connexion. Il constate que le passage est à nouveau libre (par exemple, le meuble a été déplacé de nouveau). Il transmet donc une nouvelle transition vers l'état validé (potentiellement bloqué → validé) au module Synchronisation conditionnelle, accompagnée de son propre horodatage.

Si cette seconde transition est reçue dans les 60 secondes suivant la première (c'est-à-dire à l'intérieur de la même fenêtre τ), un conflit est automatiquement détecté. Le module SYNC initie alors le processus de résolution :

Si une majorité des robots ayant observé ce chemin dans la fenêtre s'accordent sur un même

état (par exemple, 2 sur 3 robots confirment « validé »), cet état est retenu.

En cas d'égalité stricte (50/50), le module marque temporairement l'arête concerné avec un attribut *en conflit* et garde le dernier état (validé) et délègue la vérification à un opérateur humain via le module HRI.

Grâce à ce mécanisme, les décisions critiques sur l'accessibilité des connexions sont prises collectivement, tout en tenant compte de la variabilité temporelle et de la nature dynamique de l'environnement. Le système évite ainsi de figer prématurément des chemins comme bloqués ou accessibles, tout en maintenant une coordination robuste et explicable entre les agents.

A.3.6 HRI : interaction humain robot

Le module HRI est activé lorsqu'un conflit ne peut être résolu automatiquement entre les robots. Ces situations apparaissent lorsque deux ou plusieurs robots rapportent des informations contradictoires sur un même élément (par exemple, un robot considère qu'un passage est bloqué, tandis qu'un autre le juge libre), et qu'aucune majorité ne se dégage ou que l'incertitude reste trop élevée pour prendre une décision fiable. Dans ce cas, le module génère une requête explicite vers un opérateur humain, incluant le contexte du conflit, les sources des informations, et l'historique des votes ou observations. L'intervention humaine garantit une décision fiable dans les cas critiques, notamment lorsqu'une erreur de décision pourrait compromettre la mission. Le système conserve la trace de ces interventions afin d'enrichir l'historique global des modifications et potentiellement d'ajuster les politiques de vote à l'avenir. Ce module agit comme un filet de sécurité du système distribué, maintenant la robustesse et la qualité de la carte partagée dans des situations ambiguës ou exceptionnelles.

A.4 Conclusion

Cette section a présenté une extension du framework de navigation sémantique vers un système multi-robot distribué reposant sur une carte hiérarchique partagée (HSTM) et des copies locales autonomes. Cette architecture hybride équilibre l'autonomie de chaque robot et la cohérence collective du groupe, en s'appuyant sur un mécanisme de synchronisation conditionnelle, un traitement explicite des conflits et une interaction contrôlée avec un opérateur humain en cas de désaccord persistant.

L'approche multi-robot distribuée présentée dans ce chapitre offre plusieurs avantages. Tout d'abord, elle renforce considérablement la résilience du système. En répartissant la charge d'observation et de navigation entre plusieurs robots autonomes, le système est capable de tolérer des défaillances partielles : si un robot rencontre une difficulté ou une panne, les autres peuvent continuer la mission ou corriger les erreurs cartographiques associées.

L'un des atouts de ce système réside également dans sa capacité d'adaptation. Chaque robot peut réagir localement aux changements de l'environnement grâce à la mise à jour dynamique de sa carte HSTM. La synchronisation conditionnelle avec la carte globale déclenche la propagation

des informations pertinentes à l'ensemble de l'équipe, favorisant une adaptation collective à des environnements en constante évolution.

Le système peut accueillir un nombre croissant de robots sans nécessiter de refonte structurelle. Chaque agent conserve son autonomie locale, interagit selon des règles standardisées avec la carte HSTM globale, et participe aux mécanismes collectifs de mise à jour, de vote et de synchronisation. Cette capacité à faire évoluer le nombre d'agents sans rupture fonctionnelle ouvre la voie à des déploiements dans des environnements multi-robots complexes.

Dans sa configuration actuelle, Les performances globales du système notamment en termes de latence de synchronisation et de consommation mémoire liée au stockage différentiel des HSTM locales dépendront directement du nombre de robots actifs, de la fréquence des mises à jour, et de la dynamique de l'environnement. Ces paramètres feront l'objet d'une évaluation expérimentale.

En perspective, cette approche ouvre la voie à des scénarios collaboratifs plus avancés, dans lesquels les robots ne se contentent pas de partager des données cartographiques, mais coordonnent également leurs actions pour accomplir des tâches collectives complexes : transport d'objets, inspection distribuée, ou assistance conjointe à l'humain. L'intégration de modèles décisionnels partagés, d'interfaces naturelles de supervision et de stratégies d'allocation dynamique des tâches constitue autant de pistes à explorer pour étendre les capacités de coopération dans un cadre multi-agent robuste et explicable.

L'architecture actuelle du système pourrait être étendue pour intégrer des capacités de supervision distribuée à l'échelle globale en s'appuyant sur la visualisation en temps réel de la carte HSTM partagée et de la position des robots. Une telle évolution ouvrirait des perspectives intéressantes tout en nécessitant des développements complémentaires.

La mise en place d'une interface centralisée offrirait un suivi granulaire de l'activité des robots. Elle faciliterait l'analyse des déplacements et la détection des zones sous-explorées ou des congestions. Une telle fonctionnalité pourrait également autoriser un déclenchement intelligent de missions de vérification en désignant automatiquement le robot optimal pour résoudre un conflit cartographique.

Dans une version future le système pourrait gérer des réaffectations dynamiques en interrompant temporairement des tâches secondaires lorsqu'un robot se trouve en position favorable pour traiter une anomalie critique. Il pourrait aussi introduire une hiérarchisation des urgences en mobilisant des robots tiers pour les blocages majeurs tout en laissant les conflits mineurs se résoudre par consensus ou HRI.

La réalisation de ces fonctionnalités supposerait de résoudre plusieurs problèmes techniques comme la gestion des délais de communication entre les robots et l'interface centrale ou la définition de protocoles robustes pour arbitrer les décisions entre autonomie locale et supervision globale.

Ces pistes d'évolution positionnent la carte HSTM comme un outil potentiel non seulement de représentation spatiale mais aussi de coordination stratégique Elles pourraient mener à terme vers des mécanismes de planification distribuée conditionnelle intégrant l'allocation dynamique

des ressources et l'adaptation des stratégies d'intervention. Ces perspectives alignent le système avec les défis des applications en environnements dynamiques tels que la logistique ou les missions de secours.