



HAL
open science

Au delà du visible : reconstruction d'environnements par scanners laser et miroirs

Farouk Achakir

► To cite this version:

Farouk Achakir. Au delà du visible : reconstruction d'environnements par scanners laser et miroirs. Traitement du signal et de l'image [eess.SP]. Université de Picardie Jules Verne, 2021. Français. NNT : 2021AMIE0088 . tel-03969007

HAL Id: tel-03969007

<https://theses.hal.science/tel-03969007v1>

Submitted on 2 Feb 2023

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

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



Thèse de Doctorat

*Mention: Sciences pour l'ingénieur
Spécialité: Vision par ordinateur*

présentée à *l'Ecole Doctorale en Sciences en Sciences, Technologie, Santé (ED 585)*

de l'Université de Picardie Jules Verne

par

ACHAKIR FAROUK

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne

***Au-delà du visible: Reconstruction d'environnements par
scanners laser et miroirs***

Soutenue le 13 décembre 2021, après avis des rapporteurs, devant le jury d'examen :

M^{me} Marie-Odile Berger, Directrice de Recherche, INRIA Nancy	Présidente
M^{me} Sylvie Treuillet, MCF HDR, Polytech Orléan	Rapporteur
M^{me} Samia Ainouz, Professeur, l'INSA de Rouen	Rapporteur
M^{me} Marie-Odile Berger, Directrice de Recherche, INRIA Nancy	Examinatrice
M. Guillaume Caron, MCF HDR, MIS-UPJV et CNRS-AIST JRL	Examineur
M^{me} Sanaa El Fkihi, MCF HDR, ENSIAS de Rabat	Co-Directrice de thèse
M. El Mustapha Mouaddib, Professeur, UPJV	Directeur de thèse



REMERCIEMENTS

Je tiens tout d'abord à adresser mes remerciements les plus sincères à El Mustapha Mouaddib, pour avoir dirigé cette thèse et pour sa patience et ses conseils multiples tout au long de ces quatre années de thèse. Je lui exprime aussi ma gratitude pour ses grandes qualités humaines d'écoute et de compréhension qui m'ont permis de réaliser ma thèse dans les meilleures conditions

Mes remerciements vont également à ma co-directrice de thèse Sanaa El Fkihi, professeur à ENSIAS de Rabat, que je remercie pour sa disponibilité et pour toute l'aide qu'elle m'a apporté, que ce soit sur les aspects scientifiques ou techniques. Je lui exprime aussi ma gratitude pour ses relectures lors de la rédaction de mes articles et de mon manuscrit.

Je tiens aussi à remercier mon jury de thèse, à commencer par Marie-Odile Berger qui m'a fait l'honneur de le présider. Mes remerciements s'adressent aussi à mes rapporteurs Sylvie Treuillet et Samia Ainouz, pour avoir accepté de lire et d'évaluer le manuscrit et pour les commentaires qu'ils ont fait sur mon travail. Je remercie aussi les deux examinateurs Marie-Odile Berger et Guillaume Caron pour les questions et les idées intéressantes proposées lors de la soutenance.

Je n'oublie pas bien-sûr tous les autres amis et collègues du laboratoire MIS, et plus particulièrement à Eya, Eder Alejandro, Noureddine, Richardson, Yassine, Tawsif de les remercier pour tous les moments qu'on a pu passer ensemble.

Finalement, j'aimerais remercier mes parents, ainsi que toute ma famille pour leur soutien, surtout moral que j'ai pu avoir durant la préparation de la thèse.



TABLE DES MATIÈRES

Remerciements	i
Chapitre 1 : Introduction générale	1
1.1 Introduction	1
1.2 Contexte et Contributions	6
1.3 Plan du mémoire	8
Chapitre 2 : Vers une nouvelle approche de planification de vues sans modèle préalable pour une numérisation par scanner laser	11
2.1 Introduction	11
2.2 État de l’art	13
2.3 Synthèse de la méthode proposée	16
2.4 Analyse de la visibilité : La méthode projective	18
2.4.1 Détection des gaps	21
2.4.2 Classification des gaps	23
2.4.3 Correction du polygone de visibilité	25
2.4.4 Propagation des gaps	26
2.5 Stratégie d’exploration : Cellules conservatrices hiérarchiques	31
2.5.1 Génération des candidats	32
2.5.2 Scores des candidats	34
2.5.3 Sélection du meilleur candidat (Next-Best-View)	39

2.5.4	Condition d'arrêt	40
2.6	Conclusion	41
Chapitre 3 : Reconstruction d'environnement par scanner laser et miroir plan		45
3.1	Introduction	45
3.2	État de l'art	47
3.3	Synthèse de l'approche basée miroir	47
3.4	Placement automatique du miroir	51
3.5	Correction du nuage de points	56
3.5.1	Détection des mesures erronées	57
3.5.2	Correction des mesures erronées	58
3.6	Sélection de la prochaine position du miroir	59
3.7	Conclusion	61
Chapitre 4 : Résultats expérimentaux		63
4.1	Introduction	63
4.2	Simulations	64
4.2.1	Mode 1 : Simulation avec un scanner laser terrestre (TLS)	64
4.2.2	Mode 2 : Simulation du système TLS/Miroir	65
4.2.3	Mode 3 : Simulation avec un robot mobile équipé d'un scanner laser	68
4.3	Études comparatives	70
4.3.1	Mode 1 : Analyses des performances en mode hors-ligne	71
4.3.2	Mode 3 : Analyses des performances en mode temps réel	76
4.4	Expérimentation en temps réel avec un robot mobile (Mode 2)	79
4.5	Conclusion	83
Chapitre 5 : Conclusion générale et perspectives		85
5.1	Bilan des travaux réalisés	85
5.2	Quelques perspectives	88

Bibliographie	91
Liste des publications	98
Résumé	99
Abstract	99



LISTE DES FIGURES

- 1.1 **(a)** Schéma simplifié du fonctionnement d'un LIDAR. L'impulsion émise se propage dans l'espace à la vitesse de la lumière c puis elle est renvoyée au LIDAR par le réflecteur après une durée Δt . **(b)** Principe de mesure par temps de vol. **(c)** Principe de mesure par différence de phase. 2
- 1.2 Spécifications techniques du scanner laser 2D LiDAR SICK LMS300 . . 3
- 1.3 Spécifications techniques du scanner laser 3D Leica C10 3
- 1.4 **(a)** Planification des positions d'un scanner laser terrestre avec un modèle préalable (Plan 2D) de l'environnement **(b)** Exploration autonome d'un environnement avec un robot mobile équipé d'un scanner laser. Le nuage de points en bleu est superposé sur le Plan 2D à titre indicatif. . 5
- 1.5 **(a)** Reconstruction d'un buste avec un scanner laser et un miroir plan. Le scanner permet de scanner les parties qui lui sont visibles alors que le miroir permet de percevoir les parties occultées. **(b)** Reconstruction d'une colonne avec un scanner laser terrestre fixe et un robot mobile équipé d'un miroir plan. 6
- 1.6 **(a)** Extrait de la campagne de numérisation 2020 de la cathédrale Notre-Dame d'Amiens. **(b)** Une partie du nuage de points de la cathédrale. **(c)** Campagne de numérisation de l'esplanade de Hassan à Rabat **(d)** Une partie du nuage de points de l'esplanade avec la tour Hassan en haut à droite. 8

2.1	Schéma classique d'utilisation d'un planificateur de vues en mode hors-ligne (a) et en mode temps réel (b)	13
2.2	Organigramme de la méthode projective décrivant l'interaction entre ses étapes principales : la détection des gaps, la classification des gaps, la correction du polygone de visibilité et la propagation des gaps.	21
2.3	(a) Un environnement simple avec la position du scanner L_i . (b) Le nuage de points C_i avec les mesures relevées ${}^i\rho_k$. (c) La phase de détection des maxima locaux. (d) Les gaps détectés dans C_i avec la méthode proposée.	23
2.4	Exemple de détection des trois types de gaps (vrai, ouvert et fermé) à partir de la position du scanner L_i	25
2.5	La correction du polygone de visibilité permet d'identifier le gap ouvert (hachuré) et de recalculer le polygone de visibilité Ω_i	26
2.6	Le processus de la propagation des gaps G_1 dans le cas de trois positions du scanner (L_1, L_2, L_3) permet d'extraire l'ensemble des fragments invisibles ${}^1F_{inv}$ associés à la position L_1 . Après l'alignement, l'ensemble G_1 est projeté dans la fusion des polygones de visibilité $\Omega_2 \cap \Omega_3$ pour extraire ${}^1F_{inv}$	29
2.7	Le processus de propagation des gaps dans le cas de trois positions du scanner L_1, L_2 et L_3 . Les segments rouges représentent l'ensemble des fragments $G_0 = {}^1F_{inv} \cup {}^2F_{inv} \cup {}^3F_{inv}$. Les zones hachurées rouges à l'intérieur de la boîte de délimitation \mathcal{BB}_0 représentent la région occultée Ω_0^c . Les zones hachurées en vert représentent la fusion de tous les polygones de visibilité Ω_0	30

2.8	La méthode projective permet de détecter les fragments (segments rouges) et les zones occultées dans l'environnement (zone hachurée en rouge). Chaque fragment est ensuite associé à une cellule conservatrice. Les candidats de couleur bleue à l'intérieur de la cellule conservatrice associée à la région convexe présentent la même visibilité. En revanche, les candidats rouges associés à une région concave présentent des visibilités différentes.	33
2.9	Le résultat après le processus de génération des candidats appliqué au cas de la Fig. 2.8.	34
2.10	(a) Le fragment 0g_1 est visible pour le candidat c_1 . (b) Le même fragment est occulté pour c_2 du fait que $\mathcal{N}_2({}^0g_1) \neq \emptyset$	35
2.11	Un exemple d'environnement dans lequel le scanner est positionné au point L_1	37
2.12	Les scores des candidats pour différentes valeurs de λ après (a) : un (b) : deux et (c) : trois déplacements du scanner.	37
2.13	(a) : Score $s(c_i)$ ($\lambda = 0.6$) pour différentes valeurs de $\mathcal{I}_g(c_i)$ et $\mathcal{L}_p(c_i)$. (b) : Coupe profil du score $s(c_i)$ ($\lambda = 0.6$) pour une valeur fixe du gain d'informations $\mathcal{I}_g(c_i) = 5 \text{ m}^2$	38
2.14	(a) L'ensemble des candidats avant la sélection de la NBV (b) Le candidat c_7 est sélectionné pour être la NBV.	41
2.15	(a) Un exemple d'environnement avec deux positions du scanner L_1 et L_2 (b) La petite erreur d'alignement entre les deux nuages de points produit des gaps résiduels ici marqués en rouge.	42
3.1	(a) Représentation 3D du système TLS/Miroir modélisée sur Blender. (b) La représentation en 2D est une vue du dessus de l'environnement à scanner. Dans ce cas, le miroir est un segment noté m délimité par une boîte englobante notée BB_R qui représente le robot.	49

3.2	Synthèse de l'approche proposée dans ce chapitre (en rouge) et son interaction avec le planificateur de vue (en bleu) proposé dans le chapitre précédent.	50
3.3	Illustration d'un cas de figure où le placement d'un miroir permet à l'ensemble des faisceaux réfléchis de passer à l'intérieur d'un gap ${}^0g_k = ({}^0\rho_k, {}^0\rho_{k+1})$	52
3.4	(a) L'ensemble des solutions proposées comme positions admissibles du miroir (segments rouges). Certaines positions sont à l'intérieur de l'espace accessible, d'autres ne le sont pas. (b) Cas de figure où une collision a lieu entre le robot représenté par la boîte englobante BB_R et un obstacle dans l'environnement.	54
3.5	L'ensemble des positions valables pour le miroir. Pour chaque position, la boîte englobante BB_R autour du miroir est à l'intérieur de l'espace visible par le scanner laser.	55
3.6	Solution sélectionnée pour être la position suivante du miroir.	55
3.7	Représentation simplifiée de la problématique engendrée dans le cas d'un faisceau laser réfléchi par un miroir. La mesure retournée au scanner est ${}^0\rho_i'$ au lieu de la mesure correcte ${}^0\rho_i$	56
3.8	Les mesures incorrectes, ici en vert, sont placées derrière le miroir dans la zone Δ_m	57
3.9	La correction du nuage de points donnée à la figure 3.8 permet de replacer correctement les mesures en vert derrière le gap scanné par le miroir.	59
3.10	(a) Une position du miroir est générée pour chaque gap détecté dans le nuage de points. (b) la meilleure position suivante du miroir est sélectionnée par le planificateur de chemin local. Le nuage de points est enrichi par les nouvelles mesures (en vert). (c) Le processus itératif est répété pour couvrir l'ensemble de l'environnement.	61

4.1	(a) Environnement réel (lignes noires) avec les quatre positions du scanner (points jaunes). (b) Le nuage de points global C_0 avec les gaps associés aux frontières entre la zone scannée et les zones occultées.	65
4.2	L'opérateur humain procède à une numérisation partielle en déplaçant le scanner quatre fois aux points L_1, L_2, L_3 et L_4 . Le planificateur de vues calcule ensuite automatiquement les positions suivantes L_5 puis L_6 pour scanner les gaps restants.	66
4.3	(a) Détection d'un gap avec une position valable du miroir pour le scanner. (b) Le scan du miroir révèle trois nouveaux gaps g_2, g_3 et g_4 . (c) Une nouvelle position du scanner L_2 est générée après Le scan du gap g_2 avec le miroir. (d) Le nuage de points final avec les trois positions du scanner L_1, L_2 et L_3 ainsi que les trois positions du miroir.	67
4.4	Utilisation du planificateur de vues en Mode 2 dans le cas d'un environnement complexe. Les points bleus sont scannés directement avec le TLS alors que les points verts sont scannés avec miroir.	68
4.5	(a) Le processus de détection des gaps révèle deux zones occultées à proximité de la position initiale du robot L_1 . (b)-(e) Le robot explore l'ensemble de la région R_1 avant de passer à la région R_2 (f) La trajectoire finale du robot obtenue au moyen de notre stratégie d'exploration.	69
4.6	Les chemins générés par le planificateur de vues en Mode 3 dans le cas d'environnements plus complexes.	70
4.7	(a) L'esplanade de Hassan à Rabat (Maroc). (b) Nuage de points 3D de cette esplanade après 17 scans.	71
4.8	(a) Extrait 2D du nuage de points (bleu) et les 17 positions du scanner utilisées pour couvrir le centre de l'esplanade. (b) L'environnement de l'esplanade récréé avec le simulateur.	72
4.9	Résultats du calcul des candidats avec notre planificateur de vues dans l'environnement simulé du centre de l'esplanade de Hassan.	73

4.10	(a) Comparaison entre les positions générées avec notre approche (rouge) et le choix de l'opérateur humain (bleu) dans la région A_0 qui délimite le centre de l'esplanade. (b) Le taux de couverture à l'intérieur de la région A_0 après chaque déplacement du scanner.	74
4.11	Résultats du calcul des positions du scanner laser avec notre planificateur de vues pour différentes positions initiales L_1 . Le planificateur de vues permet d'assurer la couverture de l'environnement après 18 (a) ou 20 positions (b) du scanner laser.	75
4.12	Trajectoires finales générées avec chacun des planificateurs de vues sélectionnés.	76
4.13	(a) Comparaison des performances en fonction du temps de calcul. (b) Comparaison des performances en fonction de la longueur du chemin cumulé.	78
4.14	Schéma expérimental décrivant l'interaction entre le module "planificateur de vues" et le module "Robotique".	79
4.15	(a) Omron Adept MobileRobots Pioneer LX. (b) SICK LMS 300	80
4.16	(a) Position initiale du Pionner LX dans l'environnement (b) Un autre point de vue de l'environnement	80
4.17	(a) Ensemble des gaps détectés dans le nuage de points C_1 avant le processus de classification. (b) Ensemble des "vrais gaps" dans le nuage de points C_1 après le processus de classification. (c) Candidats générés par notre planificateur de vues (points bleus). (d) La NBV sélectionnée.	81
4.18	(a) Ensemble des "vrais gaps" dans le premier nuage de points C_1 et (b) dans second nuage de points C_2 après le processus de classification. (c) Détection de l'ensemble des gaps dans le nuage de points global avec la Méthode-Projective. (d) La nouvelle NBV sélectionnée par le planificateur de vues.	82
4.19	Trajectoire finale calculée en utilisant notre planificateur de vues en Mode 3.	83

-
- 5.1 **(a)** Le système FARO ScanPlan avec un TLS de la même marque **(b)**
Application mobile du Leica RTC 360 sur tablette pour le suivi du dé-
placement du TLS et l'évolution du nuage de points. 88

1.1 Introduction

Au cours des dernières décennies, diverses technologies de numérisation ont été développées notamment du fait de l'essor d'un large éventail d'applications dans le secteur de l'architecture, de l'ingénierie, la planification urbaine et la conservation du patrimoine culturel. Le scanner laser ou LIDAR (*Laser Imaging Detection and Ranging*) est utilisé pour créer des modèles numériques détaillés et précis d'environnements et permet de reproduire l'existant sous la forme d'un nuage de points dense en utilisant deux techniques distinctes.

La première consiste à envoyer une impulsion laser vers un objet de la scène à numériser. Le temps que met l'impulsion pour faire tout le parcours est appelé temps de vol ("Time Of Flight" ou "TOF") et permet de déduire la distance entre le scanner et l'objet avec précision. La deuxième technique consiste à émettre en continu des impulsions déphasées à fréquences multiples et à exploiter le décalage de phase pour mesurer les distances. Dans les deux cas, le scanner est muni d'un système optique de miroirs pivotants qui orientent les impulsions laser dans toutes les directions à très grande vitesse, chaque faisceau produit une mesure dans un ensemble discret de points communément appelé nuage de points.

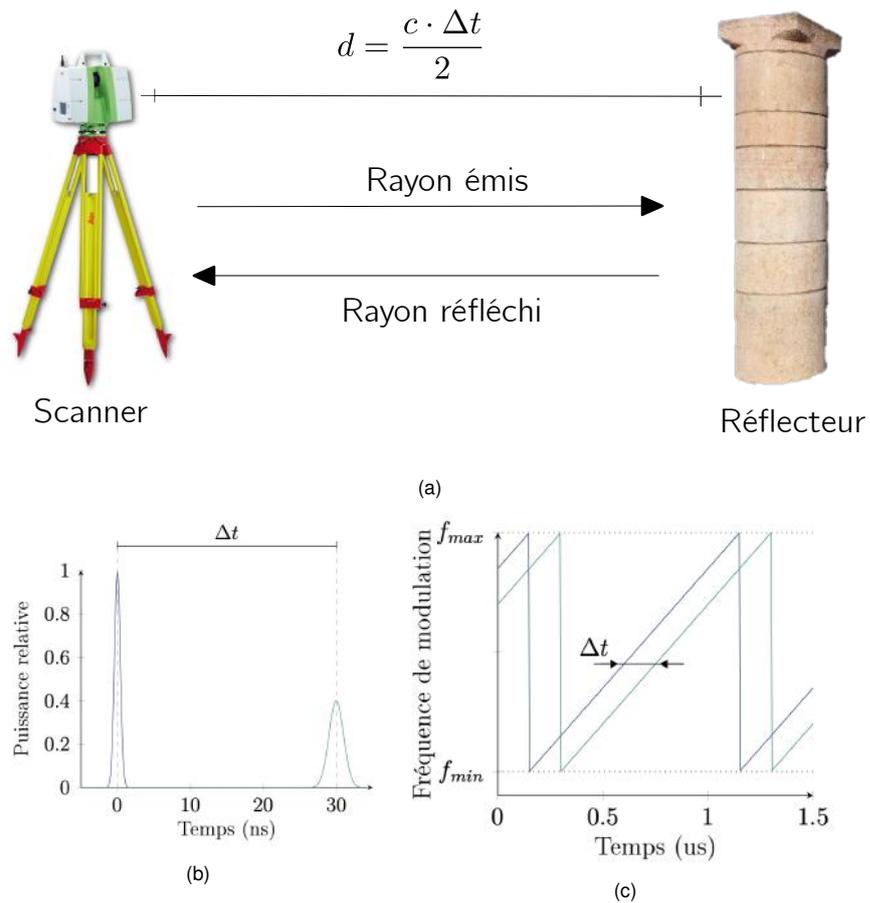


Figure 1.1 – **(a)** Schéma simplifié du fonctionnement d'un LIDAR. L'impulsion émise se propage dans l'espace à la vitesse de la lumière c puis elle est renvoyée au LIDAR par le réflecteur après une durée Δt . **(b)** Principe de mesure par temps de vol. **(c)** Principe de mesure par différence de phase.

Dans le cadre de ce travail, nous avons utilisé les données issues de deux types de capteurs LIDAR. Le premier est un capteur 2D-LiDAR SICK LMS300 à mesure par temps de vol (ToF).

Portée	de 0.5 m à 20 m
Résolution angulaire	0,5°
Angle d'ouverture	270° (horizontale)
longueur d'onde	Infrarouge (905 nm)
Domaine d'application	Indoor



Figure 1.2 – Spécifications techniques du scanner laser 2D LiDAR SICK LMS300

Le second capteur est un LIDAR 3D Leica C10 à mesure par temps de vol (ToF) adapté à une utilisation en dans de grands environnements intérieurs ainsi que pour numériser des zones en extérieur.

Portée	de 0.1 m à 300 m
Résolution angulaire	0,5°
Angle d'ouverture	360° (horizontale), 270° (verticale)
longueur d'onde	Visible (532 nm)
Domaine d'application	Indoor, Outdoor



Figure 1.3 – Spécifications techniques du scanner laser 3D Leica C10

La photogrammétrie est une approche alternative permettant à partir de mesures effectuées sur des photographies d'un objet de décrire sa forme, ses dimensions et sa position dans l'espace. Elle présente des avantages par rapport aux scanners lasers en termes de portabilité et de prix, mais aussi un certain nombre de limitations en termes de précision, de mise à l'échelle, de robustesse aux différentes textures de matériaux, etc.

L'acquisition de données, que ce soit en utilisant la lasergrammétrie ou la photogrammétrie, peut être planifiée de manière optimale pour numériser au mieux l'environnement en tenant compte de contraintes (exigences) de qualité, de temps, de coût, etc. Les pratiques courantes d'acquisition de données par scanner laser reposent sur l'expertise humaine pour la planification des emplacements du scanner et le réglage des paramètres d'acquisition à chaque emplacement sélectionné. Or, dans le cadre de la conservation du patrimoine architectural, les environnements sont souvent grands et très complexes voire en constante évolution avec les travaux de rénovation, rendant ainsi difficile, même pour des opérateurs humains expérimentés, de garantir que les nuages de points acquis couvrent entièrement l'environnement avec les niveaux de qualité requis. On peut naturellement réduire le risque de données incomplètes et peu précises en augmentant le nombre de scans effectués sur le site ; mais cette stratégie de numérisation peut entraîner des redondances dans les données rendant leur stockage et gestion plus difficiles. Par conséquent, cette masse de données importante entraîne inéluctablement une perte d'efficacité dans la gestion du projet et des retards importants dans les délais nécessaires pour numériser le site.

De tels constats ont motivé la conception et le développement d'approches (semi-) automatisées pour optimiser la numérisation par scanner laser afin d'assurer l'exhaustivité et la qualité des données requises tout en minimisant la quantité de données et les interventions des humains sur le site. Dans le domaine de la reconstruction des environnements, deux grandes classes d'approches ont émergé pour répondre au besoin de la planification des positions d'un scanner. D'abord et grâce à l'avènement de la modélisation 3D et la banalisation des logiciels de CAO/DAO, la création de plans 2D des sites à numériser est devenue accessible rendant l'approche dite avec mo-

dèle préalable (Model-Based) (Figure 1.4.a) très populaire. En effet, l'utilisation d'un plan 2D simple d'un environnement avant la visite du site permet de planifier au préalable une séquence de positions du scanner laser de prendre en compte des métriques de qualité (complétude, qualité de l'alignement, niveau de précision) dans la numérisation finale de l'environnement. Généralement utilisée en mode "hors-ligne" par un opérateur humain équipé d'un TLS, cette classe d'algorithmes nécessite intrinsèquement un modèle aussi simple soit-il de l'environnement à scanner. Or, ce modèle n'est pas systématiquement accessible et nécessite dans certains cas la collecte d'informations précises pour le réaliser, voire des compétences techniques additionnelles en modélisation 2D/3D. De telles contraintes ont motivé le développement d'approches automatisées de planification sans modèle préalable (Non-Model-Based), communément appelées algorithmes de planification de vues Next-Best-View (Figure 1.4.b). Cette classe d'approches est généralement utilisée avec des robots autonomes pour explorer de façon autonome un environnement inconnu. Dans ce cas, seules des interventions en aval pour préparer le robot et le placer correctement dans l'environnement et une intervention après la fin du processus de numérisation sont nécessaires pour obtenir un nuage de points complet du site à numériser.

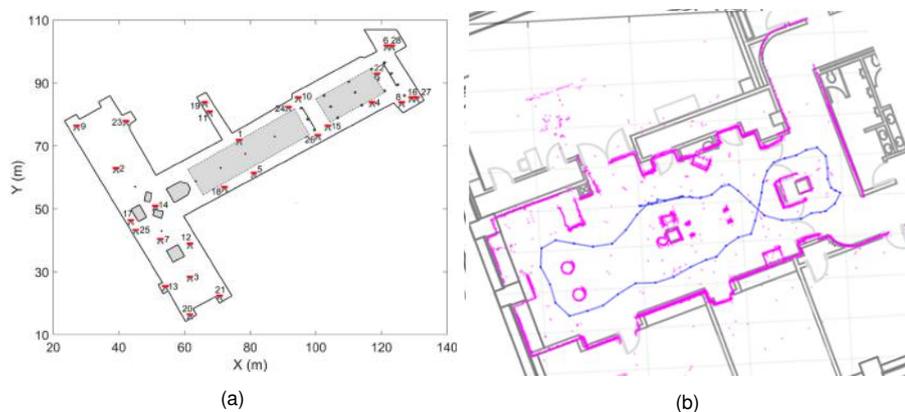


Figure 1.4 – **(a)** Planification des positions d'un scanner laser terrestre avec un modèle préalable (Plan 2D) de l'environnement **(b)** Exploration autonome d'un environnement avec un robot mobile équipé d'un scanner laser. Le nuage de points en bleu est superposé sur le Plan 2D à titre indicatif.

La majorité des approches de planification de vues avec ou sans modèle préalable utilise uniquement des scanners laser fixes ou mobiles pour numériser la scène. Pour sortir des sentiers battus, certaines approches tentent de fusionner les données des scanners laser avec d'autres technologies de numérisation (sonars, caméras) afin d'améliorer la qualité du nuage de points final. D'autres approches originales cherchent à faciliter le processus de numérisation avec des miroirs (Figure 1.5) en exploitant la propriété de réflexion totale des faisceaux laser émis par le scanner sur les surfaces spéculaires.

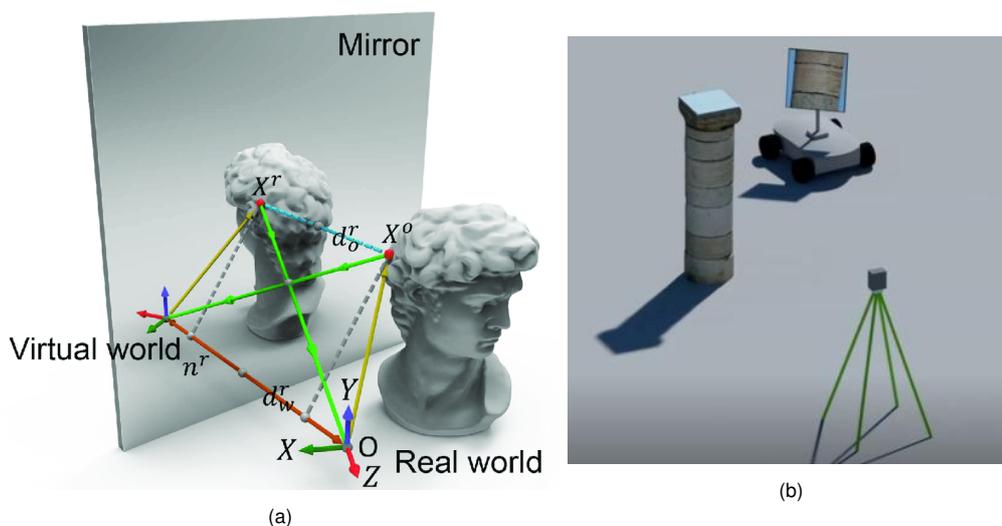


Figure 1.5 – **(a)** Reconstruction d'un buste avec un scanner laser et un miroir plan. Le scanner permet de scanner les parties qui lui sont visibles alors que le miroir permet de percevoir les parties occultées. **(b)** Reconstruction d'une colonne avec un scanner laser terrestre fixe et un robot mobile équipé d'un miroir plan.

1.2 Contexte et Contributions

Cette thèse est le fruit d'une collaboration entre le laboratoire de recherche en Informatique et Télécommunication (LRIT) de l'université Mohammed V-Rabat et le laboratoire Modélisation, Information et Systèmes (MIS) de l'université de Picardie Jules Verne à Amiens. Elle s'est déroulée dans le cadre du projet ATHAR-3D (PHC Toub-

kal). L'objectif est de développer des outils pour aider l'opérateur humain à numériser des environnements complexes avec des scanners laser terrestres (Terrestrial laser scanner/TLS) ou d'automatiser complètement le processus avec des scanners laser mobiles embarqués sur des robots terrestres (Mobile laser scanner/MLS).

En effet, au cours des dernières années, le laboratoire MIS s'est investi dans plusieurs programmes de numérisation du patrimoine architectural notamment au Maroc et en France (Figure 1.6). En France, le programme E-Cathédrale porté par le Professeur El Mustapha MOUADDIB, vise la conservation, l'amélioration de l'accessibilité, la réalisation et l'exploitation d'une maquette numérique de la cathédrale Notre-Dame d'Amiens. Au Maroc, le projet ATHAR3D, se voit comme un travail pluridisciplinaire qui implique des informaticiens, des traiteurs d'images, des roboticiens et des historiens de l'art. L'objectif est de numériser l'existant de l'esplanade de Hassan à Rabat et de compléter les parties manquantes avec des outils de la modélisation 3D.

Pour contribuer à ce chantier numérique monumental, cette thèse se donne comme objectif de réduire la complexité du processus de numérisation par l'intermédiaire d'outils informatique et robotique, en contribuant au problème de la planification des emplacements du scanner laser dans des environnements aussi complexes que ceux de la cathédrale d'Amiens ou de l'esplanade de Hassan à Rabat.

Plus spécifiquement, les principales contributions de cette thèse visent à proposer un planificateur de vues sans modèle préalable (Non-Model-Based) adapté à une numérisation par scanner laser dans des environnements larges ou complexes. Pour ce faire, le planificateur proposé est adaptatif pour permettre une utilisation en mode "hors-ligne" sur un TLS avec l'intervention d'un opérateur humain ou avec un MLS en mode "temps réel/ en ligne" pour une exploration automatique de l'environnement avec un robot mobile. De plus, une attention particulière a été apportée à l'automatisation du processus de numérisation pour solliciter le moins possible l'opérateur humain durant le processus de numérisation. Ainsi, nous proposons en plus des deux modes précédents, une troisième approche qui combine l'utilisation d'un TLS et d'un robot équipé d'un miroir plan pour réduire au maximum l'effort de déplacement du TLS qui est re-

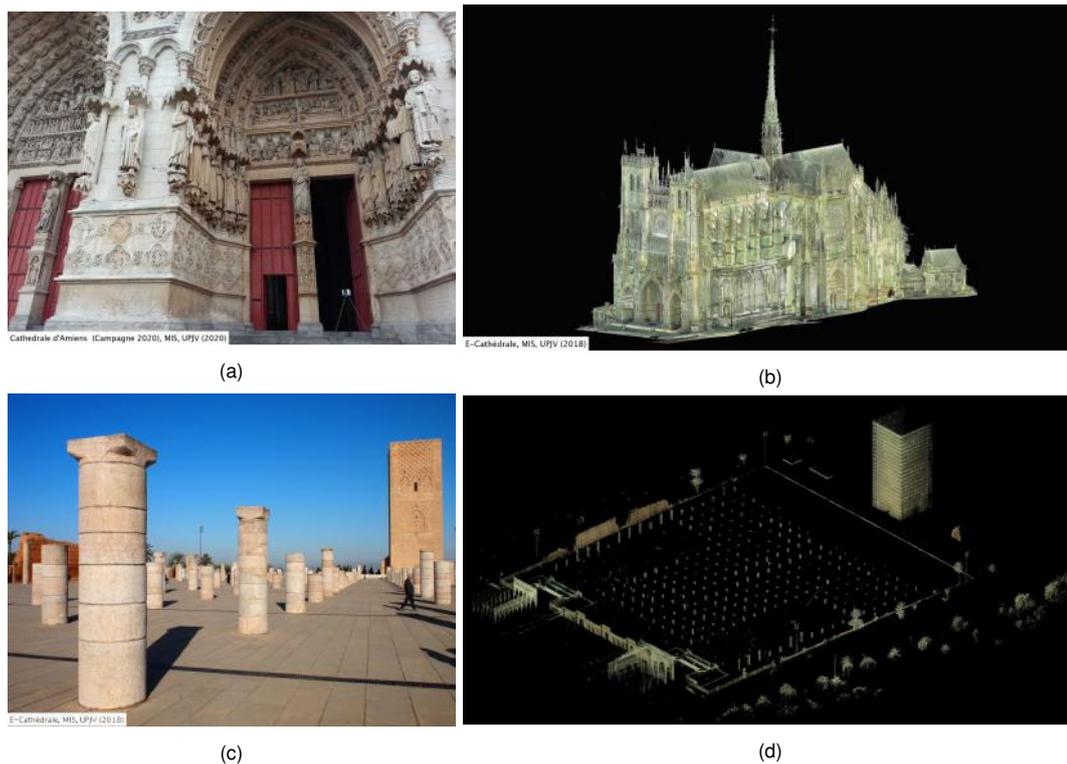


Figure 1.6 – **(a)** Extrait de la campagne de numérisation 2020 de la cathédrale Notre-Dame d’Amiens. **(b)** Une partie du nuage de points de la cathédrale. **(c)** Campagne de numérisation de l’esplanade de Hassan à Rabat **(d)** Une partie du nuage de points de l’esplanade avec la tour Hassan en haut à droite.

lativement pénible à cause de la manipulation et du positionnement des dispositifs de balayage laser.

1.3 Plan du mémoire

La suite de ce document est développée dans quatre chapitres :

Chapitre 2 : le chapitre décrit les étapes de la conception de notre planificateur de vues sans modèle préalable. Après une définition globale de la problématique de la planification de vues avec les scanners laser, un survol de l’état de l’art des principaux travaux dans ce contexte est dressé. Par la suite, une synthèse de l’approche est

proposée avant de détailler, dans la suite du chapitre, chacune des étapes du pipeline notre approche.

Chapitre 3 : Le chapitre se focalise sur l'approche proposée pour combiner l'utilisation du TLS avec un robot mobile équipé d'un miroir plan. D'abord, un état de l'art détaille les principaux travaux et l'utilisation combinée des miroirs plans avec des scanners laser. Ensuite, une synthèse de l'approche permet de décrire les étapes de fonctionnement de la méthode qui seront détaillées dans la suite du chapitre.

Chapitre 4 : le chapitre sera dédié aux résultats expérimentaux et permet d'évaluer les approches proposées dans les chapitres précédents sur un simulateur avec des données synthétiques, mais aussi sur un robot mobile. Plusieurs études comparatives ont permis d'évaluer les performances des méthodes proposées.

Enfin, le dernier chapitre de cette thèse résume les différents points clés abordés, et propose quelques perspectives de recherche.

CHAPITRE **2** **VERS UNE NOUVELLE APPROCHE DE PLANIFICATION DE VUES
SANS MODÈLE PRÉALABLE POUR UNE NUMÉRISATION PAR SCANNER LASER**

2.1 Introduction

Au cours de la dernière décennie, le recours à la numérisation d'environnements par scanner laser est devenu populaire en raison de la forte demande en modèles virtuels dans divers domaines d'application. Cette numérisation que nous supposons faite par un scanner laser, est représentée par un nuage de points et présente divers avantages, à savoir le relevé exact et fiable de l'existant, la précision de mesure pour une cartographie détaillée de l'espace physique et un accès flexible à ces informations.

Le scanner laser peut-être embarqué sur un robot mobile ou positionné par un humain dans le cas du Scanner Laser Terrestre ou Terrestrial Laser Scanner (TLS). Néanmoins, dans la majorité des cas, les environnements à numériser présentent une géométrie complexe qui rend l'utilisation des scanners laser plus difficile. En effet, il est souvent nécessaire d'effectuer plusieurs acquisitions à partir de différents emplacements afin d'obtenir une couverture maximale de l'environnement. Il en découle naturellement que le facteur clé dans l'acquisition d'un nuage de points complet est d'estimer la meilleure séquence de positions du scanner qui permet de balayer entièrement la zone numérisée de l'environnement et donc d'augmenter la couverture/complétude du modèle finale. Dans le cas d'un environnement simple, il est relativement facile de se fier à son expérience pour placer correctement le scanner et numériser toutes les zones de l'environnement. Mais, cela n'est pas toujours évident dans le cas des

environnements vastes et complexes, où il devient difficile même pour un opérateur humain expérimenté d'identifier avec précision les zones occultées qui restent à scanner et d'estimer correctement la bonne séquence de positions du scanner pour couvrir toutes les zones de l'environnement. Par conséquent, il est souvent nécessaire de recourir à des outils informatiques d'aide à la décision habituellement appelées "planificateurs de vues" pour faciliter la numérisation des environnements. De manière générale, un planificateur de vues est employé dans le cadre d'un projet de numérisation par scanner laser pour automatiser deux tâches principales, à savoir :

- L'analyse de la visibilité qui consiste à identifier avec précision les zones occultées dans un nuage de points après chaque scan.
- L'estimation des meilleures positions du scanner par le biais d'une stratégie d'exploration afin d'assurer une couverture maximale de l'environnement à numériser.

Dans ce contexte, nous proposons un planificateur de vues multi-objectif adaptatif qui peut-être utilisé comme un outil d'aide à la décision pour guider un opérateur humain et faciliter la tâche de numérisation d'un environnement inconnu ou par un robot mobile pour une exploration automatique de l'environnement. Pour ce faire, une première brique de notre planificateur de vues appelée "Méthode-Projective" permet l'analyse de la visibilité directement à partir des nuages de points fournis par le scanner sans passer par des représentations intermédiaires et avec une conception axée sur une grande précision dans la détection des zones occultées sans nécessiter d'accélération matérielle notamment dans le cas des environnements vastes et complexes. La deuxième brique de notre planificateur de vues est une nouvelle stratégie d'exploration appelée "Cellules conservatrices hiérarchiques" adaptée à une utilisation sur un TLS ou un scanner mobile tout en réduisant considérablement le temps de calcul nécessaire pour estimer la meilleure séquences des positions du scanner et cela grâce au nouveau concept de cellules conservatrices.

La section 2.2 décrit la Méthode-Projective comme première approche pour l'analyse de la visibilité dans les nuages de points. La section 2.3 détaille les différentes étapes de la stratégie d'exploration multi-objectifs : "Cellules conservatrices hiérar-

chiques" proposée pour la planification de la meilleure vue suivante du scanner laser. La section 2.4 conclut le chapitre.

2.2 État de l'art

Dans le contexte de la numérisation d'environnements par scanners laser, la classification de (Scott et al., 2003) et plus récemment celle de (Aryan et al., 2021; Lluvia et al., 2021) montrent que les travaux portant sur la planification de vues peuvent être classés en deux catégories principales. D'une part, on trouve les algorithmes de planification de vues qui se basent sur un modèle préalable de l'environnement et qui sont généralement utilisés en mode hors-ligne (Fig. 2.1a) par un opérateur humain pour guider et faciliter la tâche de la numérisation. D'autre part, on trouve les algorithmes de planification de vues sans modèle préalable, aussi appelés planificateurs de la meilleure vue suivante (Next-Best-View planners), pour un usage "temps réel" par des robots mobiles (Fig. 2.1b) pour une exploration automatique et autonome d'environnements inconnus.

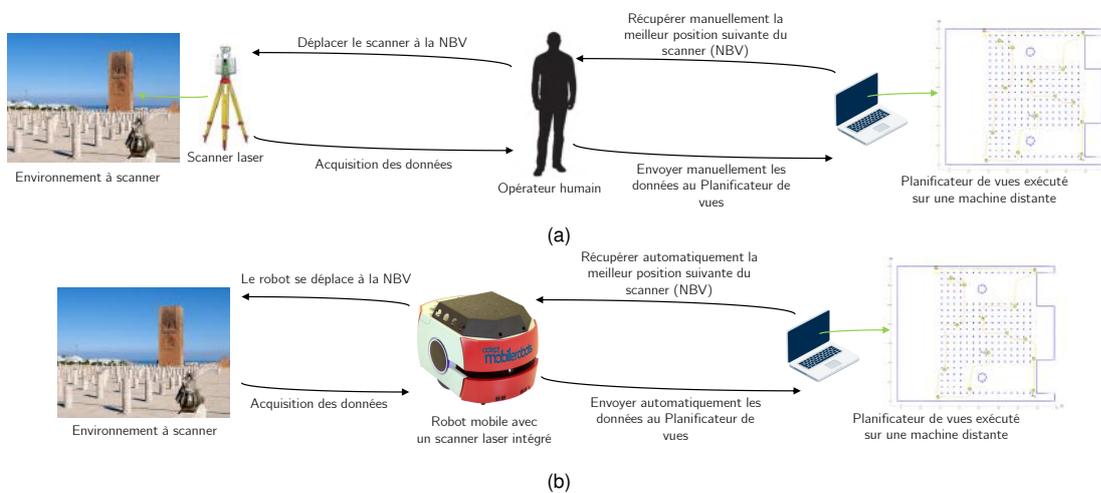


Figure 2.1 – Schéma classique d'utilisation d'un planificateur de vues en mode hors-ligne (a) et en mode temps réel (b)

La planification avec un modèle préalable a été examinée en premier lieu dans

les problèmes de la galerie d'art (O'rouke et al., 1987; Urrutia, 2000) et d'inspection de terrain (Ashok et al., 2018). Dans ce contexte, l'utilisation répandue des logiciels de CAO pour la réalisation des plans 2D (Floorplans) ou des modèles 3D des environnements à numériser a permis l'essor de cette classe d'algorithmes (Aryan et al., 2021). Ces approches sont toujours utilisées en mode hors-ligne dans le cadre d'un processus appelé P4S (Planning For Scanning) avant la visite du site à numériser, et cela pour planifier au préalable une séquence de positions du scanner laser afin d'assurer certaines métriques de qualité dans la numérisation finale de l'environnement. Plusieurs travaux (Soudarissanane & Lindenbergh, 2011; Rabbani et al., 2007; Kawashima et al., 2014; Díaz-Vilariño et al., 2018) ont été menés pour apporter une solution au P4S selon trois principales métriques de qualité : la complétude (couverture) (Biswasa et al., 2015; Son et al., 2015; Jia & Lichti, 2018), le niveau de précision (LOA) / niveau de densité (LOD) (Boehler et al., 2003; Lichti, 2007) et la précision de l'alignement entre les nuages de points (Heidari Mozaffar & Varshosaz, 2016). En outre, d'autres métriques comme le temps d'acquisition (Wujanz & Neitzel, 2016) ou d'autres facteurs qui influent sur la qualité des données, comme les propriétés des surfaces scannées (Lichti, 2008) ou l'effet de la portée et de l'angle d'incidence du laser (Soudarissanane et al., 2009, 2011), ont été considérés dans d'autres travaux. Toutefois, les plans 2D des environnements à numériser ne sont pas systématiquement accessibles et la collecte d'informations précises pour l'esquisse du plan 2D requiert généralement plus de temps ainsi que des compétences additionnelles en modélisation 2D/3D. De telles contraintes ont motivé le développement d'approches automatisées de planification sans modèle préalable, communément appelées algorithmes de planification Next-Best-View.

A contrario, la planification sans modèle préalable, également connue sous le nom de "Non-Model-Based", fait principalement référence aux algorithmes de planification de la meilleure vue suivante (Next-Best-View). Ces méthodes (Monica & Aleotti, 2018a,b; Jing & Shimada, 2018; Kawashima et al., 2014) sont principalement utilisées avec des robots autonomes qui opèrent dans un environnement inconnu sans connaissance préalable de la scène. Au cours du processus de numérisation, la carte

de l'environnement est mise à jour jusqu'à ce que la condition souhaitée de complétude/qualité du modèle final soit atteinte. La première tentative d'exploration autonome basée sur un planificateur de vues sans modèle préalable peut être attribuée à (Yamauchi, 1997). Il a proposé une approche basée sur la détection des frontières dans une grille de cellules d'occupation. Toute cellule libre adjacente à une cellule inconnue est considérée comme une cellule frontière. Ces cellules sont regroupées en régions frontalières et le robot tente de naviguer vers la frontière accessible la plus proche. Cette stratégie a connu un grand succès et de nombreux chercheurs ont ainsi développé des méthodes d'exploration basées sur les frontières. (Dornhege & Kleiner, 2013) ont étendu la méthode d'exploration de Yamauchi, basée sur les frontières 2D, aux environnements 3D en introduisant la notion de frontières 3D dans la représentation volumétrique des nuages de points "Octomap" développé par (Hornung et al., 2013). (Zhu et al., 2015) ont implémenté pour la première fois une approche basée frontières le cadre du système d'exploitation de robot ROS (Quigley et al., 2009). D'autres approches présentent des stratégies d'exploration alternatives basées sur le concept de surface frontalière (surface-based). (Senarathne & Wang, 2016) définissent une surface frontalière comme un ensemble de cellules ou au moins une des arrêtes de chaque cellule est exposée à la zone occultée de l'espace. Plus récemment (Hardouin et al., 2020) ont proposé une représentation surface-based basée sur la "TSDF" (Truncated Signed Distance Function, Curless & Levoy (1996)) afin d'identifier les zones occultées et évaluer le gain d'information et la qualité de la reconstruction 3D réalisée de manière autonome par un drone.

Tandis que la plupart des travaux utilisent une représentation volumétrique ou surfacique, (Nasir et al., 2015; Tovar et al., 2004; Murphy & Newman, 2008; Lopez-Padilla et al., 2013) utilisent directement le nuage de point pour détecter avec des approches différentes des segments appelés "gaps" qui représentent les frontières entre l'environnement scanné et les zones occultées.

Les arbres aléatoires à exploration rapide (Rapidly-exploring Random Trees / RRT) (LaValle et al., 1998) avec une stratégie d'horizon ascendant (Receding Horizon / RH) sont des alternatives à la détection des frontières, des gaps et des surfaces fronta-

lières. Ces stratégies sont appelées ainsi parce que l'horizon du robot est constamment déplacé vers l'avant, au fur et à mesure que l'environnement est numérisé. Le planificateur proposé par (Bircher et al., 2016) est considéré comme la première approche qui utilise une stratégie RRT/RH dans laquelle un arbre aléatoire RRT est calculé dans l'espace scanné (libre) et la meilleure branche est sélectionnée comme le chemin du robot. La qualité de chaque branche est déterminée par la surface de la zone occultée qui peut être explorée par le robot sur ce chemin. D'autres extensions utilisent des approches RRT-RH (Umari & Mukhopadhyay, 2017; Papachristos et al., 2017), RRT*-RH (Bircher et al., 2018; Papachristos et al., 2019) ou Theta*-RH/LazyTheta*-RH (Faria et al., 2019) des variantes de A* pour assurer la navigation autonome d'un robot dans un environnement inconnu. Pour tirer avantage de la planification RRT-RH et de la planification classique par exploration de frontière, (Selin et al., 2019; Deng et al., 2018) proposent de combiner les deux techniques. Ils utilisent la méthode d'exploration de la frontière pour définir un chemin global et la planification RRT-RH pour modifier localement ce chemin et améliorer la qualité de la numérisation.

Cette étude passe en revue les principales stratégies utilisées dans la conception d'un planificateur de vues dans le domaine de la robotique mobile. La plupart des approches étudiées suppose que le système fonctionne dans un environnement statique intérieur, les robots terrestres et aériens étant les principaux acteurs. Certains travaux de recherche tentent de résoudre le problème dans d'autres scénarios, par exemple dans des environnements sous-marins (Sutantyo et al., 2013; Shen et al., 2017)

2.3 Synthèse de la méthode proposée

Dans ce chapitre, nous proposons un planificateur de vue sans modèle préalable (Next-Best-View Planner) permettant d'accomplir la tâche d'exploration autonome (mode temps réel) si le scanner est embarqué sur un robot mobile ou de basculer en mode hors-ligne pour devenir un outil d'aide à la décision si le scanner est déplacé manuellement par un opérateur humain. La méthode proposée suppose que le scan-

ner est déplacé sur une surface plane. Ce type de numérisation est fréquent dans les environnements d'intérieur, les zones urbaines, les espaces ouverts ou encore dans certaines applications du patrimoine culturel. Plusieurs travaux récents sur la planification de vue ([Wujanz & Neitzel, 2016](#); [Jia & Lichti, 2018, 2019](#)) montrent que dans ce cas, la complexité de la planification de vue peut être réduite en considérant une coupe horizontale 2D de l'environnement 3D à hauteur du scanner. De plus, dans l'exploration autonome par robot mobile ([Wang et al., 2019](#); [Zuo et al., 2020](#); [Nasir et al., 2015](#)), l'utilisation d'une cartographie 2D performante pour la navigation associée à un scanner 3D embarqué sur le robot permet de réduire le temps d'exploration et de fournir un modèle 3D relativement complet de l'environnement.

De manière générale, un planificateur de vues sans modèle préalable adapté à une numérisation par scanner laser permet d'automatiser deux tâches principales : identifier avec précision les positions des zones occultées dans un nuage de points après chaque scan, puis estimer la meilleure position suivante (la Next-Best-View) du scanner. Pour ce faire, nous proposons deux approches.

La première approche, nommée "Méthode-Projective", permet l'analyse de la visibilité dans un nuage de points par l'identification des zones occultées au moyen d'une méthode simple appelée détection des gaps ([Nasir et al., 2015](#); [Tovar et al., 2004](#); [Murphy & Newman, 2008](#); [Lopez-Padilla et al., 2013](#)). Contrairement aux approches classiques de détection des gaps qui se limitent uniquement aux nuages de points locaux associés à une seule position du scanner à partir de plusieurs paramètres prédéfinis intrinsèquement liés à la nature de l'environnement scanné, notre méthode permet d'étendre la détection des gaps à tous les nuages de points même après le processus d'alignement, et cela, de manière adaptative sans avoir recours à des paramètres prédéfinis. De plus, les gaps sont une représentation simple, adaptée aux grands environnements et qui ne nécessite pas de temps de calcul importants à l'inverse des méthodes volumétriques ([Yamauchi, 1997](#); [González-Banos & Latombe, 2002](#); [Mobarhani et al., 2011](#)) ou des méthodes basées sur les ensembles de niveaux ([Osher & Fedkiw, 2001](#); [Subramani & Lermusiaux, 2016](#)).

La deuxième approche appelée "Cellules Conservatrices Hiérarchiques" ou Hierarchical Conservative Cells (HCC) permet d'exploiter les données fournies par la Méthode Projective lors de l'analyse de la visibilité afin de sélectionner la meilleure position suivante du scanner. À la différence des algorithmes de planification de vue sans modèle préalable qui génèrent un grand nombre de candidats soit sur les frontières pour maximiser la visibilité des candidats (Yamauchi, 1997; Mobarhani et al., 2011; Gao et al., 2018), soit en échantillonnant les candidats sur l'ensemble de la région préalablement scannée (Wang et al., 2018; Foulds, 2012), notre planificateur utilise des régions spécifiques appelées "Cellules conservatrices" pour réduire considérablement le nombre de candidats. Les deux approches proposées sont combinées pour obtenir le meilleur compromis entre la complétude du modèle final et le temps d'exploration en particulier dans le cas des environnements vastes et complexes.

2.4 Analyse de la visibilité : La méthode projective

Le processus d'analyse de la visibilité consiste à produire une représentation numérique simple de l'environnement qui permet d'identifier de manière précise à la fois les obstacles et les frontières entre les régions préalablement scannées et les zones occultées par les obstacles.

Dans le cas d'une numérisation par scanner laser, il est souvent nécessaire d'effectuer plusieurs scans à partir de plusieurs positions afin de couvrir l'ensemble des surfaces visibles de la scène. Chaque scanner situé à un emplacement L_i produit un nuage de points $C_i = \{^i\rho_0, ^i\rho_1, \dots, ^i\rho_k\}$ dans un repère local \mathcal{F}_i , où $^i\rho_j \in \mathbf{R}^2$ désigne les coordonnées cartésiennes de la j -ième mesure pour chaque faisceau laser envoyé par le scanner. Par la suite, l'alignement permet de déterminer la transformation 0T_i entre chaque repère local \mathcal{F}_i et un repère global \mathcal{F}_0 commun à tous les nuages de points ${}^0T_i \cdot C_i$.

En fonction des scénarios d'utilisation du scanner laser, la vitesse du processus d'alignement peut être primordiale si le capteur est embarqué sur un robot mobile

ou déplacé manuellement par un opérateur humain dans le cas d'une station TLS. Dans le premier cas d'utilisation, des méthodes locales de recalage telles que Iterative-Closest-Point (Besl & McKay, 1992) ou Normal-Distributions-Transform (Magnusson, 2009) peuvent être utilisées pour converger plus rapidement. Dans le deuxième cas d'utilisation, lors de l'utilisation d'une station TLS, la précision de l'alignement est plus importante et par conséquent l'utilisation d'une méthode globale telle que Coherent-Point-Drift (CPD) (Myronenko & Song, 2010) est plus judicieuse.

Le processus d'alignement permet de générer un nuage de points global noté C_0 dans un repère global \mathcal{F}_0 commun à tous les nuages de points ${}^0T_i \cdot C_i$ tel que :

$$C_0 = \bigcup_{i=1}^N {}^0T_i \cdot C_i = \bigcup_{i=1}^N {}^0C_i \quad (2.1)$$

où N désigne le nombre de positions du scanner.

Chaque nuage de points 0C_i peut-être représenté sous la forme d'une région sans obstacle appelée "Polygone de visibilité local" noté ${}^0\Omega_i$. La fusion de tous les polygones de visibilité locaux, permet de définir la région visible de l'environnement aussi appelée polygone de visibilité global Ω_0 tel que :

$$\Omega_0 = \bigcup_{i=1}^N {}^0T_i \cdot \Omega_i = \bigcup_{i=1}^N {}^0\Omega_i \quad (2.2)$$

D'une part, le nuage de points global C_0 permet d'identifier avec précision les positions des obstacles dans un seul et même référentiel \mathcal{F}_0 . D'autre part, le polygone de visibilité global Ω_0 permet de quantifier la superficie de la région scannée de l'environnement et de déterminer une région où il est garanti que le scanner peut se déplacer sans risque de collision avec les obstacles fixes. Néanmoins, aucune des deux représentations ne permet de définir avec précision les position des frontières dans l'environnement scanné. En effet, le contour du polygone Ω_0 est constitué d'un ensemble de segments qui peuvent être à la fois des portions des surfaces d'objets préalablement scannés ou des frontières qui délimitent une séparation entre les zones

scannées et les zones occultées.

L'analyse de la visibilité dans nuage de points après l'alignement s'avère être une problématique importante et rares sont les approches qui permettent de la résoudre de manière précise en combinant uniquement les nuages de points et les polygones de visibilité. En effet, la majorité des approches propose de convertir les nuages de points en grille d'occupation et d'utiliser des techniques de lancer de rayons afin de simplifier la détection des frontières au détriment d'un temps de calcul important notamment dans le cas des environnements larges et complexes.

Nous proposons dans cette section, une méthode qui permet de détecter de manière très précise les frontières directement dans un nuage de points après le processus d'alignement. Pour y parvenir, notre approche consiste à faire une analyse locale de la visibilité dans chaque nuage de points C_i . Par la suite, l'alignement et les polygones de visibilité sont utilisés pour propager la visibilité locale préalablement calculée à l'ensemble du nuage de points C_0 . Nous appelons cette méthode : la méthode projective. Cette dernière est articulée autour de trois étapes principales :

1. La détection des gaps : Le but de cette étape est de détecter les gaps dans chaque nuage de points locaux C_i . Dans ce contexte, une nouvelle approche adaptative pour la détection des gaps est proposée.
2. La correction du polygone de visibilité : Cette étape permet d'analyser les effets de la portée maximale du scanner ainsi que sa résolution angulaire sur la complétude d'un nuage de points et son polygone de visibilité. Une correction est apportée si nécessaire pour remédier à l'incomplétude du nuage de points due à la discontinuité des mesures du scanner dans certaines situations.
3. Propagation des gaps : Cette étape permet de combiner les polygones de visibilité pour propager les gaps détectés préalablement dans les nuages de points locaux C_i à l'ensemble du nuage de points C_0 .

La figure 2.2 illustre visuellement la méthode proposée pour l'analyse de la visi-

bilité. Les différentes étapes de cette méthode sont détaillées dans les sections suivantes.

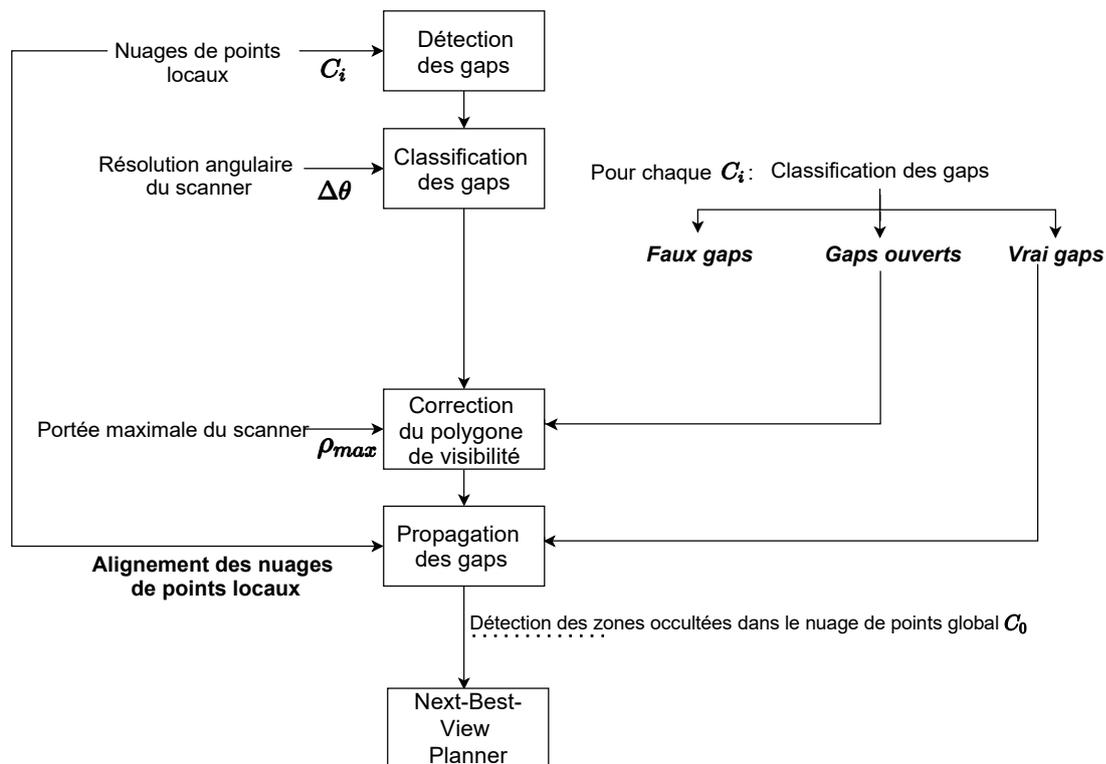


Figure 2.2 – Organigramme de la méthode projective décrivant l'interaction entre ses étapes principales : la détection des gaps, la classification des gaps, la correction du polygone de visibilité et la propagation des gaps.

2.4.1 Détection des gaps

La géométrie computationnelle, la méthode des ensembles de niveaux (Level-Set) ou les méthodes volumétriques de lancer de rayons (Ray-Tracing) sont des approches permettant de résoudre le problème de l'analyse de la visibilité dans des nuages de points locaux ou globaux. Toutefois, le temps de calcul pour extraire les frontières devient rapidement important dans le cas des environnements larges et complexes. Ainsi, les approches basées sur la détection des gaps fournissent une représentation géométrique simple des frontières appelée "gaps" adaptée à tout type d'environnement.

Le processus de la détection des gaps est défini dans les travaux antérieurs (Nasir et al., 2015; Tovar et al., 2004; Murphy & Newman, 2008; Lopez-Padilla et al., 2013) comme étant la détection d'un segment de ligne appelé "gap" noté ${}^i g_k = ({}^i \rho_k, {}^i \rho_{k+1})$ qui relie deux mesures ${}^i \rho_k$ et ${}^i \rho_{k+1}$ dans un nuage de points 2D local C_i et indique l'existence d'une frontière entre une région visible et une région occultée par des obstacles. Les extrémités d'un gap ${}^i g_k$ vérifient l'équation suivante :

$$|{}^i \rho_k, {}^i \rho_{k+1}| \geq \delta_{min} \quad (2.3)$$

Où δ_{min} est défini comme un seuil positif qui désigne la longueur minimale supposée pour un gap.

La principale difficulté dans l'utilisation des méthodes de détection des gaps réside dans la sélection d'un seuil optimal pour la détection de tous les gaps dans le nuage de points. On peut choisir empiriquement une valeur statique pour ce seuil, qui restera constante tout au long du processus de détection des gaps. Cependant, en dépit des efforts de l'utilisateur, le seuil statique ne permet pas toujours de détecter avec précision tous les gaps. En effet, le seuil δ_{min} dépend fortement des caractéristiques de l'environnement (largeur des portes ou des couloirs dans un environnement intérieur, ou espace entre les bâtiments dans un environnement urbain).

Pour éviter d'utiliser un seuil statique, nous proposons de détecter les maxima locaux dans la variation de la distance euclidienne entre toutes les mesures consécutives d'un nuage de points 2D. Chaque maximum local à une position k est utilisé pour calculer un segment de ligne (écart) qui relie deux mesures ${}^i \rho_k$ et ${}^i \rho_{k+1}$.

La figure 2.3a représente un scanner à la position L_i au milieu d'un environnement simple pour illustrer le processus de détection des gaps avec la méthode proposée. Dans un premier temps, le scanner produit un nuage de points C_i . Puis, quatre maxima locaux sont détectés aux positions $k = \{10, 131, 136, 165\}$ dans le nuage de points C_i (Fig. 2.3c). Pour chaque position k , la mesure ${}^i \rho_k$ est connectée à ${}^i \rho_{k+1}$ pour former un gap noté ${}^i g_k$. Comme le montre la figure 2.3d, l'approche pro-

posée permet de détecter dans le nuage de points C_i un ensemble final de gaps noté $G_i = \{^i g_{10}, ^i g_{131}, ^i g_{136}, ^i g_{164}\}$.

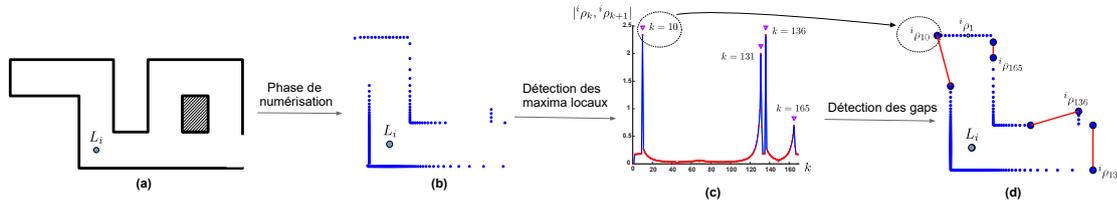


Figure 2.3 – (a) Un environnement simple avec la position du scanner L_i . (b) Le nuage de points C_i avec les mesures relevées $^i \rho_k$. (c) La phase de détection des maxima locaux. (d) Les gaps détectés dans C_i avec la méthode proposée.

2.4.2 Classification des gaps

Le processus de détection des gaps produit un ensemble de gaps (voir Fig. 2.3d) lesquels sont supposés indiquer la présence d'une frontière entre la partie scannée et les régions occultées dans un nuage de points. Toutefois, ce processus est étroitement lié aux caractéristiques intrinsèques du scanner. Ainsi, la portée maximale du scanner ρ_{max} et la résolution angulaire $\Delta\theta$ entraînent la formation de discontinuités dans les mesures de nuage de points qui produisent après le processus de détection des gaps des fausses frontières similaires à de véritables gaps.

Le processus de classification des écarts permet de procéder à l'analyse de chaque écart et de le classer dans l'une des trois catégories suivantes (Fig. 2.4) : "vrai gap", "gap ouvert" et "faux gap". La première catégorie définit les "vrais gaps" et fait référence à tous les gaps qui sont alignés avec la position du scanner L_i . Par conséquent, un gap $^i g_k = (^i \rho_k, ^i \rho_{k+1})$ est ajouté à cette catégorie si l'équation 2.4 est vérifiée :

$$d_{\perp}(L_i, ^i g_k) \simeq 0 \quad (2.4)$$

où $d_{\perp}(L_i, ^i g_k)$ désigne la projection orthogonale de la position du scanner L_i sur le support du gap $^i g_k$.

Par ailleurs, la résolution angulaire $\Delta\theta$ du scanner laser a une influence importante sur le processus de détection des gaps. En effet, plus les objets sont éloignés de la position du scanner, plus la distance entre les mesures consécutives du nuage de points sera grande, ce qui a pour conséquence de créer des "faux gaps" (Voir Fig. 2.4). Dans ce cas, les gaps ne sont pas alignés avec la position L_i et les deux points délimitant "faux gap" sont consécutifs de sorte que l'angle ${}^i\theta_{k,k+1} = \angle {}^i\rho_k L_i {}^i\rho_{k+1}$ est égal à la résolution angulaire du scanner $\Delta\theta$. Ainsi, les contraintes qui permettent de détecter un "faux gap" ${}^i g_k$ sont formulées comme suit :

$$\begin{cases} d_{\perp}(L_i, {}^i g_k) > 0 \\ {}^i\theta_{k,k+1} \simeq \Delta\theta \end{cases} \quad (2.5)$$

En outre, dans certains cas, il arrive que tous les obstacles dans certaines régions de l'environnement soient au-delà de la portée maximale du scanner ρ_{max} . Par conséquent, le scanner ne génère aucune mesure pour ces zones et cette discontinuité dans les mesures produit un "gap ouvert" lors du processus de détection des gaps (Voir Fig. 2.4). Dans ce cas, l'angle ${}^i\theta_{k,k+1}$ entre les deux points délimitant le "gap ouvert" est supérieur à $\Delta\theta$. Les contraintes qui permettent de détecter les "gaps ouverts" sont :

$$\begin{cases} d_{\perp}(L_i, {}^i g_k) > 0 \\ {}^i\theta_{k,k+1} > \Delta\theta \end{cases} \quad (2.6)$$

Après le processus de la classification des gaps, les "faux gaps" sont supprimés du nuage de points, les "gaps ouverts" sont utilisés lors du processus de la correction des polygones de visibilité (sous-section 2.4.3) et les " vrais gaps " sont utilisés lors de l'étape de la propagation des gaps (sous-section 2.4.4).

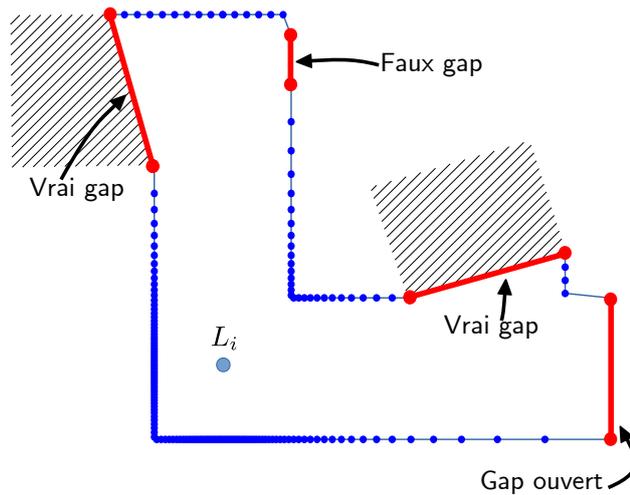


Figure 2.4 – Exemple de détection des trois types de gaps (vrai, ouvert et fermé) à partir de la position du scanner L_i .

2.4.3 Correction du polygone de visibilité

Chaque nuage de points local C_i est une représentation discrète de l'environnement qui entoure le scanner. Connecter les points successifs dans ce nuage de points permet de produire un contour qui délimite une région dépourvue d'obstacles, appelée polygone de visibilité et notée Ω_i . On calcule généralement ce polygone pour déterminer les zones où le capteur peut se déplacer librement. En conséquence, la détermination correcte du polygone de visibilité est essentielle pour garantir que le capteur ne percute aucun obstacle lors de ses déplacements. Toutefois, la présence d'un ou de plusieurs "gaps ouverts" dans le nuage de points C_i après le processus de classification indique que le nuage de points est incomplet, ce qui génère un faux polygone de visibilité Ω_0 .

Le processus de correction du polygone de visibilité consiste à identifier les extrémités de chaque "gap ouvert" (${}^i\rho_k, {}^i\rho_{k+1}$). Sachant que tous les faisceaux laser qui traversent un gap ouvert ne fournissent aucune mesure à cause de l'absence d'obstacles, nous simulons la rotation du faisceau laser depuis le point ${}^i\rho_k$ jusqu'au point ${}^i\rho_{k+1}$ en substituant à chaque fois dans les nuages de points l'absence des mesures

par les valeurs simulées.

La figure 2.5 illustre le polygone de visibilité Ω_i après le processus de correction. Le "gap ouvert" (Voir Fig. 2.4) dans le nuage de points est transformé en deux "vrais gaps" permettant ainsi d'explorer une nouvelle zone de l'environnement.

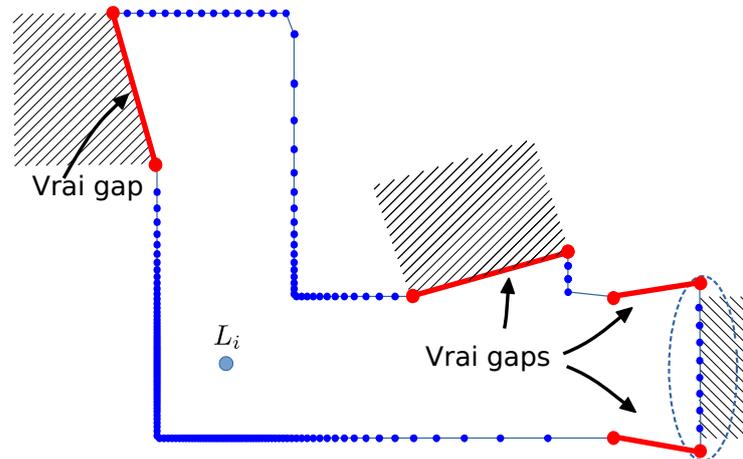


Figure 2.5 – La correction du polygone de visibilité permet d'identifier le gap ouvert (hachuré) et de recalculer le polygone de visibilité Ω_i .

2.4.4 Propagation des gaps

En général, dans un environnement complexe, il est nécessaire de déplacer le scanner plusieurs fois afin de générer un ensemble de nuages de points C_i qui sont par la suite alignés et fusionnés en un seul nuage de points C_0 . Étant donné qu'après l'alignement il n'est plus possible d'identifier les points consécutifs dans le nuage de points C_0 , les méthodes basées sur la détection des gaps deviennent inopérantes. En outre, la détection des gaps dans chaque nuage de points local C_i ne permet pas de déterminer les gaps dans le nuage de points global C_0 après le processus d'alignement. En effet, un gap détecté depuis une position L_i n'est pas forcément présent dans C_0 car il peut-être visible depuis une autre position $L_{j \neq i}$.

Afin d'utiliser notre approche de détection des gaps après le processus d'aligne-

ment, il faut évaluer l'impact de toutes les positions du scanner sur chaque gap. Nous proposons une approche que nous appelons "propagation des gaps" pour générer un ensemble des fragments de gaps G_0 dans le nuage de points global C_0 en utilisant les transformations géométriques 0T_i précédemment définies dans l'équation 2.2 et les opérateurs booléens de la topologie des polygones : (OR) noté \cup , (AND) noté \cap , le complément booléen (NOT). Le complément du polygone Ω_i est noté Ω_i^c et l'opérateur (XOR) noté \oplus . Ces opérateurs sont appliqués aux deux entités géométriques qui interviennent dans le processus à savoir : les polygones de visibilité Ω_i et les ensembles de "vrais gaps" notés G_i .

L'idée fondamentale à la base de notre méthode est de faire une projection de chaque ensemble de segments de "vrais gaps" $G_i = \{g_1; g_2; \dots; g_k; \dots\}$ d'une position L_i dans les polygones de visibilité $\Omega_{j \neq i}$ des autres positions du scanner $L_{j \neq i}$. Puis, nous identifions pour chaque gap la partie du segment qui est visible (*définition 1*) ou invisible (*définition 2*) pour les autres positions du scanner $L_{j \neq i}$. Nous appelons "fragments" les parties visibles et invisibles des gaps.

Définition 1: Fragments visibles

Les fragments visibles d'un ensemble G_i associé à une position L_i du scanner sont les parties des gaps qui sont visibles depuis au moins une autre position $L_{j \neq i}$. Cet ensemble noté ${}^iF_{vis}$ est formalisé par l'équation suivante :

$$\begin{aligned} {}^iF_{vis} &= \bigcup_{j, j \neq i} ({}^0T_i \cdot G_i \cap {}^0T_j \cdot \Omega_j) \\ &= \bigcup_{j, j \neq i} ({}^0G_i \cap {}^0\Omega_j) \\ &= {}^0G_i \cap \bigcup_{j, j \neq i} {}^0\Omega_j \end{aligned} \quad (2.7)$$

Avec ${}^0G_i = {}^0T_i \cdot G_i$ et ${}^0\Omega_i = {}^0T_i \cdot \Omega_i$

Définition 2: Fragments invisibles

Les fragments invisibles d'un ensemble G_i associé à une position L_i sont les parties des gaps qui sont invisibles depuis toutes les autres positions $L_{j \neq i}$. Cet ensemble est noté ${}^i F_{inv}$ et est calculé à l'aide de l'équation suivante :

$$\begin{aligned} {}^i F_{inv} &= {}^0 T_i \cdot G_i \oplus {}^i F_{vis} \\ &= {}^0 G_i \oplus {}^i F_{vis} \end{aligned} \quad (2.8)$$

La Fig. 2.6 synthétise la propagation des gaps de l'ensemble G_1 des positions du scanner L_1, L_2 et L_3 . L'ensemble final ${}^1 F_{inv}$ est représenté comme un sous-ensemble de G_1 après le processus de propagation.

Dans un premier temps, l'ensemble des gaps G_1 associé à la position L_1 et les polygones de visibilité Ω_2 et Ω_3 associés respectivement aux positions L_2 et L_3 du scanner sont alignés dans un même référentiel \mathcal{F}_0 . Ensuite, la *définition 1* est utilisée pour extraire l'ensemble des fragments visibles ${}^1 F_{vis}$ de G_1 et la *définition 2* est utilisée pour extraire l'ensemble des fragments invisibles ${}^1 F_{inv}$.

Le même processus est répété pour les deux ensembles de gaps G_2 et G_3 associés respectivement aux positions L_2 et L_3 . La figure 2.7 présente le résultat du processus de propagation des gaps avec les mêmes positions du scanner L_1, L_2 et L_3 en appliquant l'équation 2.9. Le processus de propagation des gaps permet d'extraire les ensembles de fragments invisibles ${}^1 F_{inv}$, ${}^2 F_{inv}$ et ${}^3 F_{inv}$ associés à chacune des trois positions du scanner. Enfin, la fusion de ces ensembles permet de trouver G_0 (segments rouges) qui représente les fragments dans le nuage de points global C_0 .

Ainsi, l'ensemble des fragments G_0 dans le nuage de points global C_0 peut être défini par la fusion de tous les fragments invisibles ${}^i F_{inv}$:

$$G_0 = \bigcup_{i=1}^N {}^i F_{inv} \quad (2.9)$$

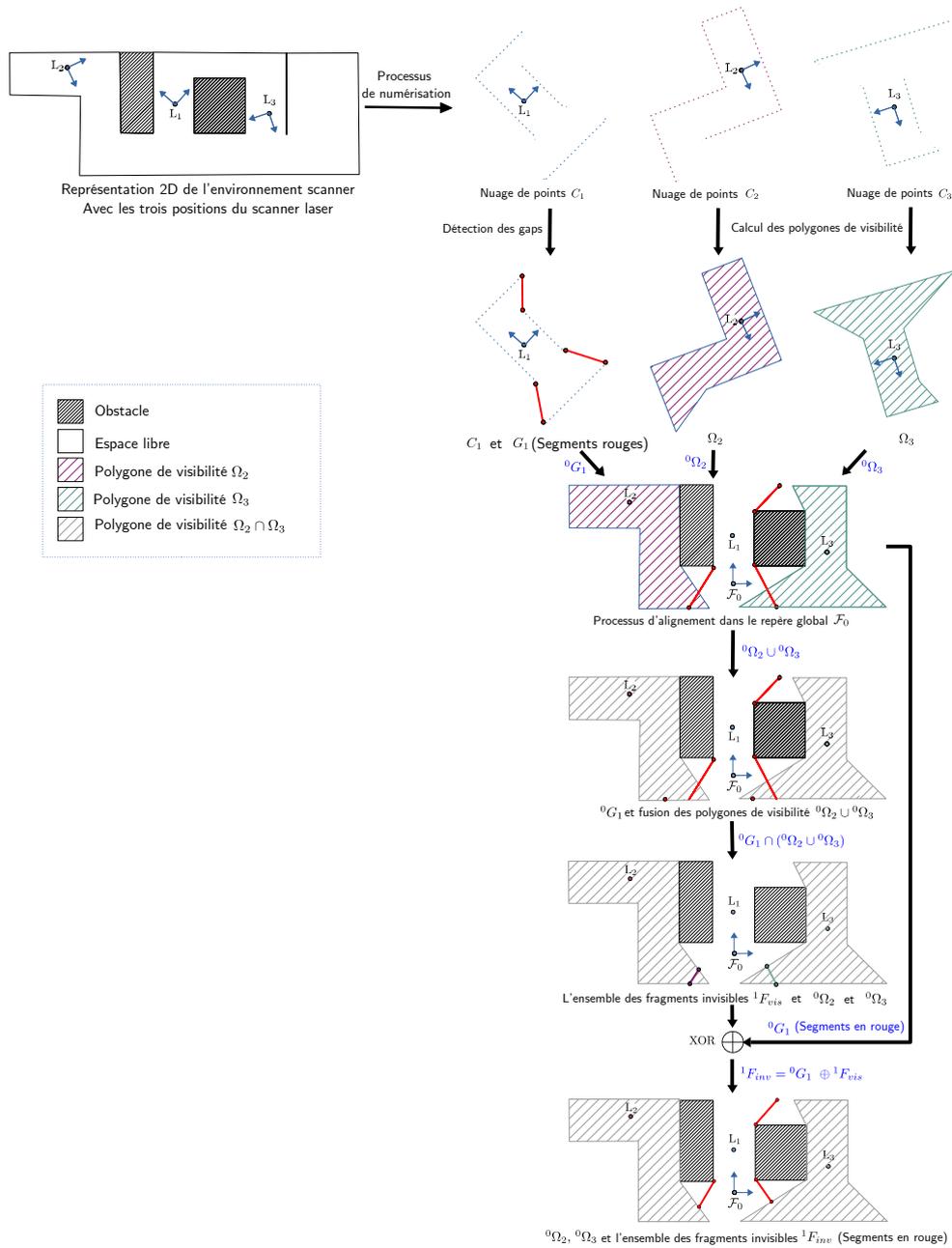


Figure 2.6 – Le processus de la propagation des gaps G_1 dans le cas de trois positions du scanner (L_1 , L_2 , L_3) permet d'extraire l'ensemble des fragments invisibles ${}^1F_{inv}$ associés à la position L_1 . Après l'alignement, l'ensemble G_1 est projeté dans la fusion des polygones de visibilité $\Omega_2 \cap \Omega_3$ pour extraire ${}^1F_{inv}$.

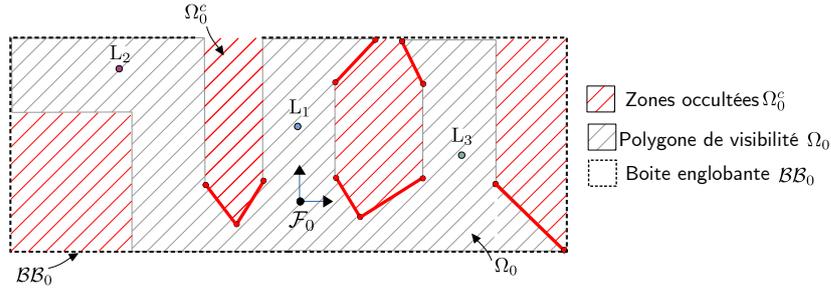


Figure 2.7 – Le processus de propagation des gaps dans le cas de trois positions du scanner L_1 , L_2 et L_3 . Les segments rouges représentent l'ensemble des fragments $G_0 = {}^1F_{inv} \cup {}^2F_{inv} \cup {}^3F_{inv}$. Les zones hachurées rouges à l'intérieur de la boîte de délimitation BB_0 représentent la région occultée Ω_0^c . Les zones hachurées en vert représentent la fusion de tous les polygones de visibilité Ω_0 .

Par ailleurs, la fusion de tous les polygones de visibilité représente la région scannée précédemment définie sous le nom de polygone de visibilité global Ω_0 où :

$$\Omega_0 = \bigcup_{i=1}^N {}^0\Omega_i \quad (2.10)$$

Cette région Ω_0 est délimitée par une zone rectangulaire appelée boîte englobante BB_0 .

Dès lors, la région non scannée Ω_0^c est définie comme le complément de Ω_0 à l'intérieur de la boîte englobante BB_0 où :

$$\Omega_0^c = \left(\bigcup_{i=1}^N {}^0\Omega_i \right)^c \quad (2.11)$$

La Méthode-Projective permet d'extraire l'ensemble des zones occultées dans un nuage de points global et de les représenter par des gaps. Par la suite, l'approche présentée dans la section suivante appelée "Cellules conservatrices hiérarchiques" ou Hierarchical Conservative Cells (HCC) permet d'exploiter la position des gaps pour déterminer la meilleure position suivante (Next-Best-View) du scanner laser.

2.5 Stratégie d'exploration : Cellules conservatrices hiérarchiques

La méthode projective détaillée précédemment permet l'analyse de la visibilité dans un nuage de points défini comme la fusion des données C_i acquises à partir de plusieurs positions L_i du scanner. La visibilité est alors représentée par l'ensemble des fragments de gaps G_0 , l'espace libre noté Ω_0 , et la région non scannée de l'environnement notée Ω_0^c .

La frontière entre l'espace libre Ω_0 et la région non balayée Ω_0^c est définie par l'ensemble des fragments G_0 . Un fragment n'étant rien d'autre qu'un segment qui relie deux mesures dans le nuage de points, il ne représente donc aucun objet réel dans l'environnement réel. Par conséquent, les seules ouvertures possibles pour scanner la région Ω_0^c sont l'ensemble des fragments $G_0 = \{^0g_1, ^0g_2, \dots, ^0g_k\}$ calculés par le biais de la méthode projective (équation 2.9). Sur la base de cette observation, nous proposons une approche permettant de maximiser la visibilité du scanner à travers les fragments G_0 et d'optimiser le mouvement du scanner.

Pour ce faire, nous proposons une approche appelée Stratégie d'exploration par cellules conservatrices hiérarchiques (Hierarchical Conservative Cells Exploration Strategy ou HCC) qui consiste en trois étapes majeures :

- Génération d'un ensemble de candidats potentiels pour la prochaine position du scanner (Next-Best-View).
- Évaluation de chaque candidat en fonction de critères préalablement sélectionnés.
- Sélection du meilleur candidat pour la prochaine position du scanner.

Dans la suite de cette section, nous allons détailler ces différentes étapes.

2.5.1 Génération des candidats

Un opérateur humain ou un robot équipé d'un scanner ne peut se déplacer vers la position suivante que si cette dernière se trouve dans l'espace libre défini préalablement comme Ω_0 . Un certain nombre d'approches (Connolly, 1985; Vásquez-Gómez et al., 2009; Dornhege & Kleiner, 2013) utilisent cette contrainte pour générer dans l'espace libre un ensemble de candidats associés aux gaps préalablement détectés, et cela afin de sélectionner la meilleure position en fonction d'un ou plusieurs critères. Cependant, après chaque déplacement du scanner, les gaps changent et il est nécessaire de générer à chaque fois un grand nombre de candidats dans tout l'espace libre. Cette opération est très coûteuse en terme de temps de calcul d'autant plus que l'espace libre devient de plus en plus grand au fur et à mesure de la numérisation, en particulier si l'environnement est large ou nécessite plusieurs déplacements du scanner.

De ce fait, la conception de notre approche de planification de vues est principalement axée sur la réduction du nombre de candidats générés en travaillant uniquement sur des régions spécifiques de l'espace libre que nous appelons cellules conservatrices. Dans ce contexte, nous aurons besoin des deux définitions suivantes :

Définition 3: La ligne d'inflexion

Une inflexion est une variation du signe de la courbure du contour du polygone de visibilité Ω_0 . La ligne d'inflexion est une ligne prolongée à partir de l'inflexion.

Définition 4: La cellule conservatrice

La cellule conservatrice associée à un fragment est définie comme la région délimitée par le fragment lui-même et les lignes d'inflexion prolongées depuis chacune de ses extrémités jusqu'à ce que les deux lignes se rencontrent.

La particularité d'une cellule conservatrice est que la visibilité d'un capteur donné

reste maximale et constante, quelle que soit sa position à l'intérieur de celle-ci. Par conséquent, il suffit d'associer à chaque fragment de G_0 une cellule conservatrice et de placer le scanner à l'intérieur de celle-ci pour garantir une visibilité maximale du capteur. De plus, cela permet de réduire le nombre de candidats associés à chaque fragment. Comme le montre la Fig. 2.8, la même visibilité sera obtenue par les candidats bleus, indépendamment de leurs positions à l'intérieur de la cellule conservatrice. Il n'est donc plus nécessaire de générer plusieurs candidats dans l'espace libre, mais une seule position associée à chaque cellule conservatrice.

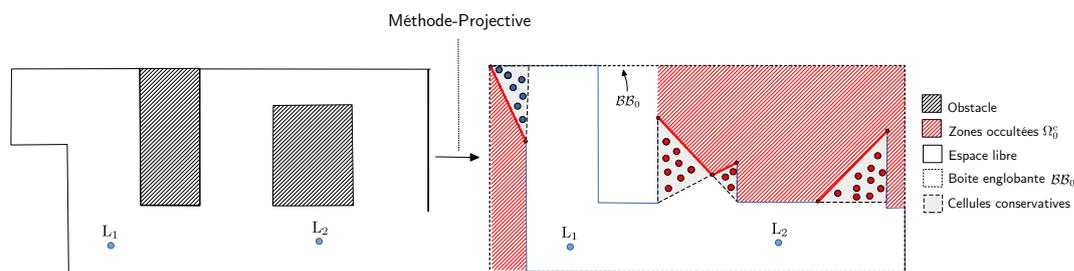


Figure 2.8 – La méthode projective permet de détecter les fragments (segments rouges) et les zones occultées dans l'environnement (zone hachurée en rouge). Chaque fragment est ensuite associé à une cellule conservatrice. Les candidats de couleur bleue à l'intérieur de la cellule conservatrice associée à la région convexe présentent la même visibilité. En revanche, les candidats rouges associés à une région concave présentent des visibilités différentes.

À noter que la *définition 4* est seulement valable dans le cas de fragments associés à une région convexe. Comme le montre la Fig. 2.8, la visibilité des candidats en rouge change en fonction de leurs positions, cela confirme que l'utilisation de cellules conservatrices ne peut plus être envisagée dans le cas de régions concaves. En conséquence, nous avons introduit la *proposition 1* qui permet de sélectionner une position spécifique où la visibilité du capteur est maximisée pour un fragment associé à une région concave.

Proposition 1

La visibilité au travers d'un fragment associé à une région concave est maximisée

sur une de ses extrémités et minimisée sur l'autre.

Preuve. Une région concave associée à un fragment peut être subdivisée en au moins deux sous-régions convexes. Chacune des sous-régions convexes peut être associée à une cellule conservatrice représentée par un candidat. L'intersection de toutes les cellules conservatrices correspond à l'une des extrémités du fragment. Ainsi, le candidat placé sur cette extrémité permet d'observer le plus grand nombre de sous-régions convexes et par conséquent maximiser la visibilité au travers du fragment.

Dans la figure 2.9, nous présentons un exemple du processus de génération de candidats. Dans le cas d'un fragment associé à une région convexe, un seul candidat (bleu) est généré à l'intérieur de la cellule conservatrice. Dans le second cas où la région concave est délimitée par trois fragments, un candidat (rouge) est généré par fragment sur chacune des extrémités qui maximisent la visibilité selon la proposition 1.

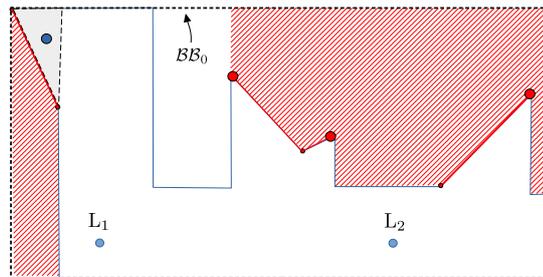


Figure 2.9 – Le résultat après le processus de génération des candidats appliqué au cas de la Fig. 2.8.

2.5.2 Scores des candidats

A la fin du processus de génération des candidats, chaque candidat noté c_i est associé à un fragment. Il faut alors vérifier s'il permet de scanner des régions associées à d'autres fragments. En pareil cas, il est nécessaire de vérifier qu'aucun obstacle dans l'environnement ne peut obstruer la visibilité du candidat. Pour cela, le champ de vision de chaque candidat c_i associé à un gap 0g_k est subdivisé en un premier niveau de visibilité noté $\mathcal{N}_i({}^0g_k)$ et un second niveau de visibilité noté $\mathcal{B}_i({}^0g_k)$.

Le premier niveau de visibilité $\mathcal{N}_i({}^0g_k)$ est représenté par la zone triangulaire comprise entre la position du candidat c_i et le fragment 0g_k . Elle conditionne le fait qu'un fragment soit directement visible ou non depuis c_i . Ainsi, un fragment 0g_k est invisible depuis la position candidate c_i si :

$$\mathcal{N}_i({}^0g_k) \cap \Omega_0^c \neq \emptyset \quad (2.12)$$

Lorsqu'un fragment ${}^0g_k \in G_0$ est visible, le candidat c_i a recours au deuxième niveau de visibilité pour pouvoir observer la zone située au-delà du fragment. Ce niveau de visibilité est noté $\mathcal{B}_i({}^0g_k)$ et permet de mesurer la surface de la région $\mathcal{R}({}^0g_k, \mathcal{B}\mathcal{B}_0)$ comprise entre le fragment 0g_k et la boîte englobante $\mathcal{B}\mathcal{B}_0$ du nuage de points C_0 , tel que :

$$\mathcal{B}_i({}^0g_k) = \text{Area}(\mathcal{R}({}^0g_k, \mathcal{B}\mathcal{B}_0) \cap \Omega_0^c) \quad (2.13)$$

La figure 2.10 illustre deux candidats c_1 et c_2 à l'intérieur d'une même cellule conservatrice associée à un fragment 0g_2 . Les deux candidats utilisent leurs premiers niveaux de visibilité pour vérifier si un autre fragment 0g_1 est visible depuis leurs positions. Le fragment 0g_1 est visible pour c_1 mais occulté pour c_2 car $\mathcal{N}_2({}^0g_1) \neq \emptyset$

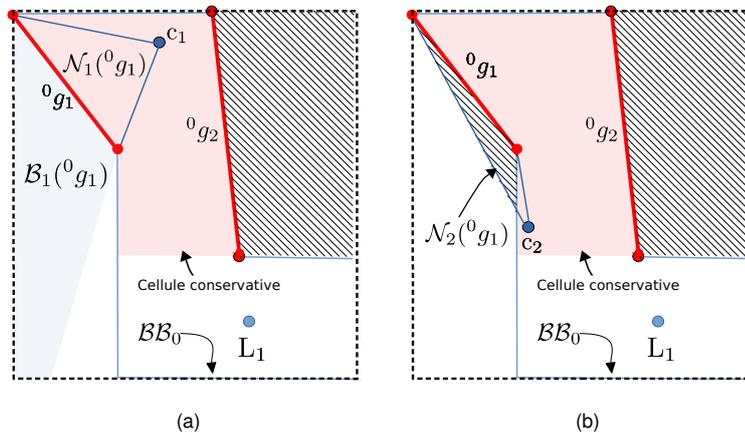


Figure 2.10 – (a) Le fragment 0g_1 est visible pour le candidat c_1 . (b) Le même fragment est occulté pour c_2 du fait que $\mathcal{N}_2({}^0g_1) \neq \emptyset$.

Étant donné que le candidat c_i peut observer plus d'un fragment dans l'ensemble G_0 , le gain d'information $\mathcal{I}_g(c_i)$ de c_i est défini comme suit :

$$\mathcal{I}_g(c_i) = \sum_{k=1}^{\mathcal{V}(c_i)} \mathcal{B}_i^0(g_k) \quad (2.14)$$

où $\mathcal{V}(c_i)$ désigne le nombre de fragments visibles.

Outre le critère de gain d'information, la distance entre la position de chaque candidat et la position du scanner actuelle est un facteur important pour optimiser le temps nécessaire au processus de numérisation. Dès lors, le critère de la longueur du chemin sans collision $\mathcal{L}_{\mathcal{P}}(c_i)$ entre la position du scanner actuel et le candidat c_i est également intégré dans la définition du score final s_i associé à chaque candidat c_i :

$$s(c_i) = \lambda \cdot \mathcal{I}_g(c_i) + (1 - \lambda) \cdot \mathcal{L}_{\mathcal{P}}^{-1}(c_i) \quad (2.15)$$

La constante positive $\lambda \in [0, 1]$ est utilisée pour pondérer le coût du déplacement par rapport au gain d'information.

Afin de sélectionner une valeur de λ qui assure le meilleur compromis entre la longueur finale du chemin et le nombre minimum de candidats à sélectionner pour couvrir tout l'environnement, on propose une technique empirique avec des scénarios différents (bureaux intérieurs, champs ouverts extérieurs,...) pour analyser les performances de notre processus de planification. Pour illustrer cette méthode, nous proposons le scénario simple d'un scanner laser positionné au point L_1 dans un environnement 2D similaire à un espace de bureaux (Fig. 2.11).

La méthode projective permet ensuite de détecter trois fragments face auxquels trois candidats c_1 , c_2 et c_3 sont positionnés (Fig. 2.12a). Pour chaque candidat, nous avons tracé la variation des scores $s(c_1)$, $s(c_2)$, $s(c_3)$ en fonction de λ . Ces graphiques montrent que le score du candidat c_1 est le plus élevé pour toutes les valeurs de λ .

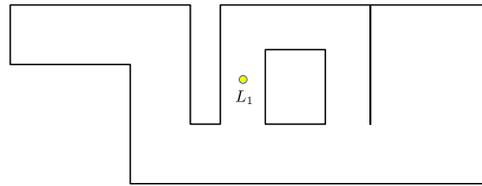


Figure 2.11 – Un exemple d’environnement dans lequel le scanner est positionné au point L_1 .

Il est donc sélectionné pour devenir la seconde position du scanner L_2 . À partir de la position L_2 , le même processus permet de sélectionner le candidat c_1 pour devenir la troisième position du scanner L_3 (Fig. 2.12b).

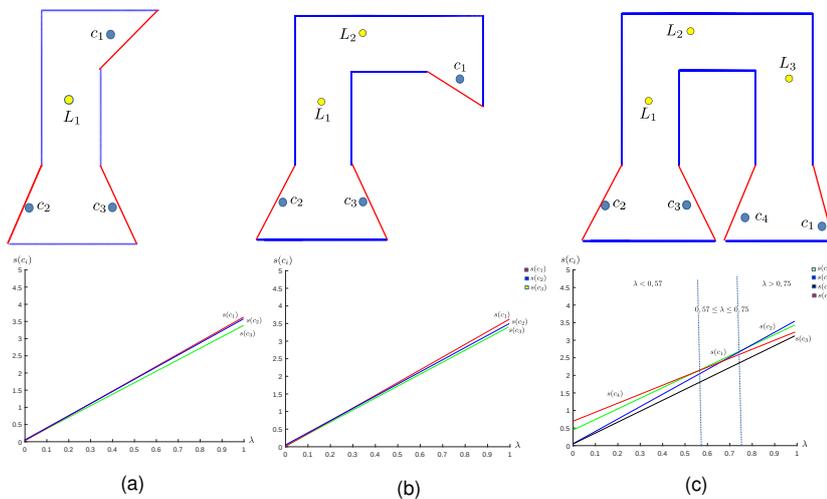


Figure 2.12 – Les scores des candidats pour différentes valeurs de λ après (a) : un (b) : deux et (c) : trois déplacements du scanner.

En revanche, le troisième scan (position L_3) (Fig. 2.12c) révèle quatre candidats et aucun d’entre eux ne présente un score maximal pour toutes les valeurs de λ . On distingue dans ce cas trois situations différentes :

- $\lambda < 0.57$: Le score $s(c_4)$ est le plus élevé et le candidat c_4 est sélectionné pour être la position de L_4 .
- $0.57 \leq \lambda \leq 0.75$: Le score $s(c_1)$ est le plus élevé et le candidat c_1 est sélectionné pour être la position L_4 .

— $0.75 < \lambda$: Le score $s(c_2)$ est le plus élevé et le candidat c_2 est sélectionné pour être la position L_4 .

Nous constatons que le chemin le plus court pour explorer l'environnement est généré avec une valeur de $0.57 \leq \lambda \leq 0.75$. Ainsi, après avoir effectué d'autres simulations similaires sur des environnements différents, nous avons retenu que la valeur $\lambda = 0.6$ parce qu'elle offre un bon compromis entre la longueur du chemin final et le nombre de positions du scanner laser.

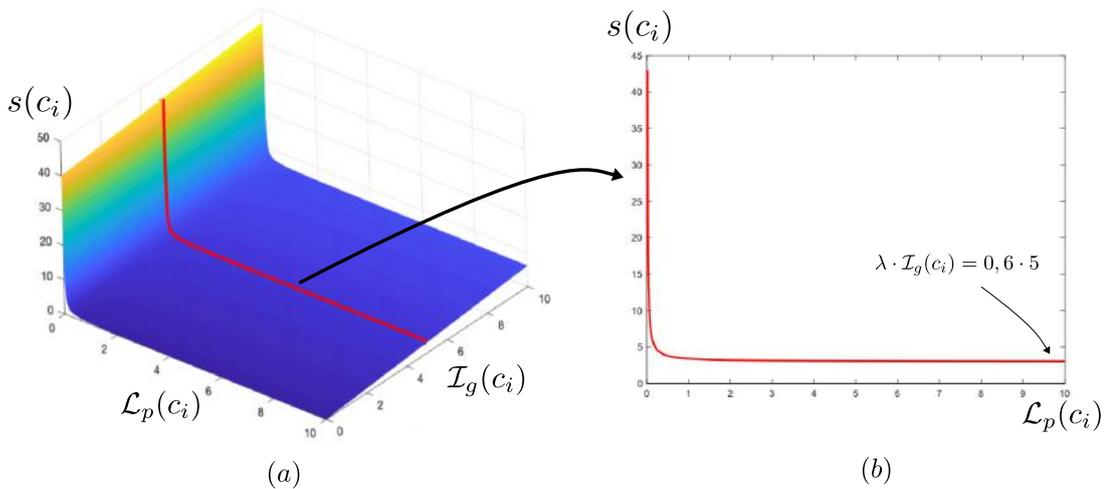


Figure 2.13 – **(a)** : Score $s(c_i)$ ($\lambda = 0.6$) pour différentes valeurs de $\mathcal{I}_g(c_i)$ et $\mathcal{L}_p(c_i)$. **(b)** : Coupe profil du score $s(c_i)$ ($\lambda = 0.6$) pour une valeur fixe du gain d'informations $\mathcal{I}_g(c_i) = 5 \text{ m}^2$

Nous avons également tracé l'évolution du score $s(c_i)$ ($\lambda = 0.6$) pour différentes valeurs de $\mathcal{I}_g(c_i)$ et $\mathcal{L}_p(c_i)$. Le résultat comme illustré sur la figure 2.13.a montre que le score $s(c_i)$ dépend fortement de la distance $\mathcal{L}_p(c_i)$. Afin de mieux illustrer ça, nous avons effectué une coupe de profil (figure 2.13.b) pour une valeur fixe du gain d'informations $\mathcal{I}_g(c_i) = 5 \text{ m}^2$. Nous remarquons que le score augmente rapidement dans le cas si le candidat est très proche de la position courante du scanner laser ($\mathcal{L}_p(c_i) < 1\text{m}$). Dans le cas contraire d'un candidat plus loin de la position courante du scanner laser ($\mathcal{L}_p(c_i) > 1\text{m}$), le score tend rapidement vers une valeur égale à $\lambda \cdot \mathcal{I}_g(c_i)$.

Par conséquent, deux cas de figure peuvent intervenir dans la suite du processus. Dans le premier cas, l'étape suivante de génération des candidats produira un ensemble de positions candidates proches et loin de la position courante du scanner laser. Les candidats proches auront donc des scores très élevés et leurs sélections seront par la suite privilégiées. Dans le second cas, aucun candidat très proche de la position du scanner n'est généré et le choix parmi les candidats loin va dépendre fortement du gain d'informations qu'ils peuvent apporter à la numérisation de l'environnement.

2.5.3 Sélection du meilleur candidat (Next-Best-View)

Après le processus de génération et d'évaluation des candidats, l'algorithme de planification de vues du HCC se charge de sélectionner la meilleure vue suivante. Ainsi, chaque fragment est représenté par un seul candidat sélectionné en fonction de la région associée (convexe ou concave) :

- **Fragment associé à une région convexe** : Tous les candidats à l'intérieur de la cellule conservatrice du fragment ont la même visibilité. Ainsi, un seul candidat à l'intérieur de la cellule peut être choisi pour représenter tous les autres. Le candidat sélectionné par notre stratégie est situé à $\mathcal{L}(c_i)$, le centre de la cellule conservatrice (*définition 4*).
- **Fragment associé à une région concave** : Le candidat retenu est placé au point $\mathcal{L}(c_i)$, l'extrémité du fragment qui maximise la visibilité selon la *proposition 1*.

La localisation $\mathcal{L}(c_i)$ de chaque candidat c_i permet de calculer son score associé $s(c_i)$ ainsi que le niveau du candidat qui correspond au nombre de fragments visibles $\mathcal{L}(c_i)$ depuis $\mathcal{V}(c_i)$. Les candidats sont ensuite hiérarchisés en fonction de leur score et de leur niveau.

La stratégie de sélection proposée consiste à identifier un candidat c_i qui présente un bon compromis entre un score élevé et le maximum de fragments visibles. Un score

élevé signifie que le candidat permet un gain d'information important et un coût de déplacement faible depuis la position courante du scanner. Par ailleurs, un candidat c_i avec un grand nombre de fragments visibles $\mathcal{V}(c_i)$ permet de scanner en même temps plusieurs zones occultées. Par exemple, comme le montre la Fig. 2.14a, c_3 est un candidat avec un score très élevé, mais il ne peut observer qu'un seul fragment $\mathcal{V}(c_1) = 1$. Au contraire, le fait d'observer plusieurs fragments, comme c'est le cas pour le candidat c_6 , ne garantit pas nécessairement un score plus élevé. Pour y remédier, un bonus est accordé à chaque candidat c_i en fonction du nombre de fragments qu'il observe $\mathcal{V}(c_i)$. La disparité des scores des candidats est évaluée par la moyenne des différences successives carrées (MSSD) noté $\delta^2(s)$. Ensuite, chaque candidat observant plus d'un fragment ($\mathcal{V}(c_i) > 1$) voit son score augmenter de $(\mathcal{V}(c_i) - 1) \cdot \delta^2(s)$. Ainsi, un candidat c_i observant un seul fragment ($\mathcal{V}(c_i) = 1$) reste au même score initial.

Enfin, la meilleure position suivante (Next-Best-View) du scanner positionné à L_i est définie selon la formule suivante :

$$NBV_{i+1} = \mathcal{L}(c_k \mid k = \underset{x}{\operatorname{argmax}} \{s(c_x) + \delta^2(s) \cdot (\mathcal{V}(c_x) - 1)\}) \quad (2.16)$$

La figure 2.14 montre un ensemble de candidats pour lesquels la MSSD est égale à 5,22. Le candidat c_7 est sélectionné comme vue suivante, car il permet un bon équilibre entre le score $s(c_7)$ et le nombre de fragments visibles : $\mathcal{V}(c_7) = 3$.

2.5.4 Condition d'arrêt

Le processus d'exploration est interrompu dès lors que la longueur de tous les fragments de l'ensemble G_0 devient inférieure à un seuil prédéfini. Si ce critère est vérifié après le processus de propagation des gaps, alors la numérisation de l'environnement est terminée. Autrement, la stratégie d'exploration passe à l'étape suivante de la génération des candidats.

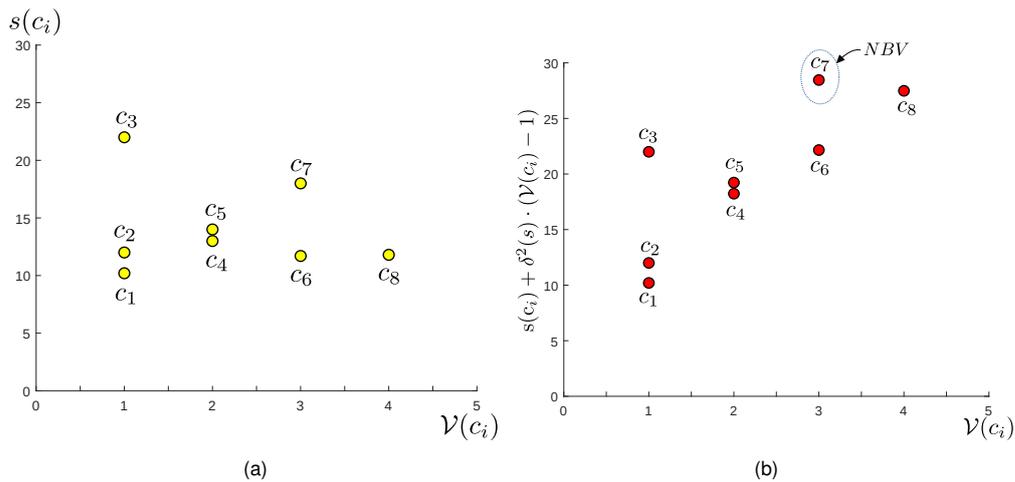


Figure 2.14 – (a) L'ensemble des candidats avant la sélection de la NBV (b) Le candidat c_7 est sélectionné pour être la NBV.

La condition de d'arrêt permet également de traiter l'effet des petites erreurs d'alignement sur le processus de projection des gaps. En général, en mode hors-ligne, l'opérateur humain se charge de vérifier visuellement si l'alignement est correct avant de d'intégrer les nouveaux scans dans le planificateur de vues. Néanmoins, dans le cas d'un robot mobile, l'alignement des scans peut mal se passer à cause des mouvements brusques ou rapides et être peu précis comme le montre la figure 2.15. La conséquence d'une petite erreur d'alignement des nuages de points locaux se traduit par l'apparition de petits gaps résiduels. Toutefois, le seuil fixé par la condition d'arrêt permet d'outrepasser les gaps résiduels qui sont généralement plus petits que la taille minimale fixée pour les fragments. On peut également limiter la vitesse linéaire et angulaire du robot pour réduire l'erreur d'alignement des nuages de points afin de créer une carte plus précise de l'environnement.

2.6 Conclusion

Dans ce chapitre, nous avons proposé une approche sans modèle préalable pour résoudre le problème de la planification de vues. Notre pipeline "Méthode-Projective +

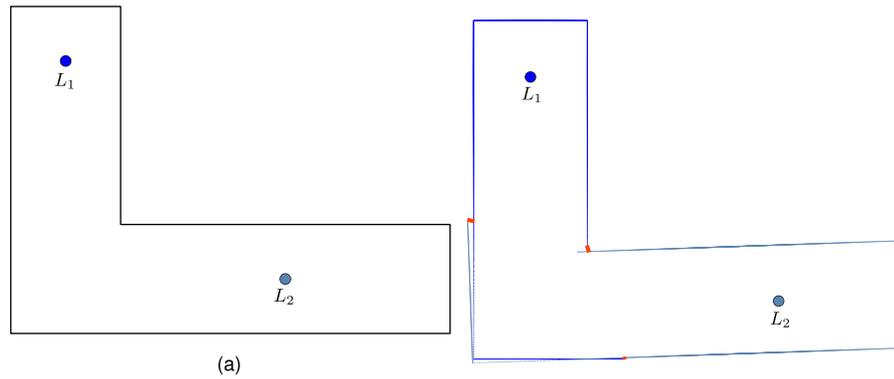


Figure 2.15 – **(a)** Un exemple d'environnement avec deux positions du scanner L_1 et L_2 **(b)** La petite erreur d'alignement entre les deux nuages de points produit des gaps résiduels ici marqués en rouge.

Cellules Conservatrices Hiérarchique", permet l'analyse de la visibilité et le placement optimal du scanner dans le cas d'un TLS en mode hors-ligne ou d'un scanner mobile en mode temps réel. La conception de notre approche a été axée sur deux objectifs majeurs. Le premier est de pouvoir effectuer une analyse de visibilité dans des environnements vastes et complexes, et ce directement à partir des nuages de points fournis par le scanner et sans passer par des représentations intermédiaires. Cet objectif est atteint avec la Méthode-Projective qui offre une grande précision sans nécessiter d'accélération matérielle comme c'est le cas avec les méthodes volumétriques. Le deuxième objectif est réalisé par l'approche "Cellules Conservatrice Hiérarchique" en réduisant considérablement le nombre de candidats pour la sélection de la meilleure position suivante du scanner et cela grâce au concept de cellules conservatrices.

Le planificateur de vue en mode temps réel permet d'automatiser complètement la numérisation d'un environnement avec uniquement des interventions amont pour préparer le robot et le placer correctement dans l'environnement, et une intervention en aval après la fin du processus de numérisation pour récupérer les données du modèle final de l'environnement scanner. Dans le cas du mode hors-ligne, le planificateur de vue peut-être utilisé comme un outil d'aide à la décision pour orienter un opérateur humain dans le placement d'un scanner laser terrestre. Par conséquent, l'intervention de l'opérateur humain est récurrente dans le processus de numérisation pour dépla-

cer le TLS dans l'environnement, une tâche certes minimisée par le planificateur de vues mais qui reste relativement pénible au vu du poids des dispositifs de balayage laser. Le chapitre suivant détaille une approche basée sur l'utilisation d'un robot mobile équipé d'un miroir pour réduire au maximum les déplacements du TLS et ainsi réduire considérablement l'effort fourni par l'opérateur humain.

Les performances du pipeline proposé dans ce chapitre seront détaillées dans le chapitre "Résultats expérimentaux". Notre stratégie a été évaluée dans des environnements simulés et réels. Dans le cas du mode temps réel avec un scanner mobile embarqué sur une plateforme robotique où l'approche fournit des résultats satisfaisants en terme de taux de couverture dans un environnement intérieur (locaux du laboratoire MIS). En outre, la méthode a également été comparée avec d'autres planificateurs de vues, ainsi qu'avec un opérateur humain expérimenté dans le cas d'un environnement extérieur complexe et de grande taille (L'esplanade de Hassan à Rabat).

3.1 Introduction

Dans le contexte de la numérisation des environnements, les scanners laser conviennent parfaitement aux applications qui requièrent une grande précision, comme l'architecture, la cartographie urbaine et la topographie. Néanmoins, le processus de numérisation par scanner laser dans des environnements larges et complexes est parfois difficile et délicat même pour un opérateur expérimenté.

L'automatisation du processus de numérisation par le biais des planificateurs de vues permet de faciliter le travail en proposant un panel d'outils pour automatiser des tâches ardues telles que l'analyse de la visibilité ou le choix des positions optimales du scanner laser. De plus, l'utilisation des scanners embarqués sur des robots mobiles permet parfois d'automatiser complètement la numérisation d'un environnement avec uniquement des interventions en amont pour préparer le robot et le placer correctement dans l'environnement et une intervention en aval après la fin du processus de numérisation pour récupérer les données du modèle final de l'environnement scanner.

Il en découle naturellement que l'intérêt pour les scanners laser a augmenté au fur et à mesure que les algorithmes de planification de vues ont pris de l'ampleur avec des approches de plus en plus rapides et robustes. Cette démocratisation des dispositifs de balayage laser particulièrement appréciés pour leur précision dans plusieurs types d'environnements a contribué à révéler certaines limites liées à leurs principes de fonctionnement. Toutefois, l'utilisation d'un faisceau laser a ses défauts. En effet, la

présence de surfaces spéculaires et notamment des miroirs dans l'environnement va causer la réflexion totale du faisceau laser et la position du point scanné sera erronée. D'autres limites du scanner laser concernent les surfaces transparentes comme les vitraux, les vitres et les fenêtres dans les espaces de bureau par exemple qui produisent également des mesures erronées.

Nous proposons dans ce chapitre de tirer avantage de la réflexion des faisceaux lasers sur les miroirs afin de faciliter le déplacement d'un scanner terrestre par un opérateur humain. L'intérêt principal de cette approche est de compléter notre planificateur de vue en mode "hors-ligne" précédemment détaillé dans le chapitre 2. En effet, dans ce mode, l'opérateur humain déplace un TLS, une tâche certes minimisée par le nombre optimal des positions fournies par le planificateur de vues, mais qui reste relativement pénible au vu du poids des dispositifs de balayage laser. Dans le cas du mode "temps réel", le planificateur de vues est utilisé avec un robot mobile et on considère que le processus est déjà totalement automatisé et ne nécessite que très peu d'intervention de la part de l'opérateur humain. Pour ce faire, nous proposons d'intégrer une brique supplémentaire à notre planificateur de vues décrit dans le chapitre 2 en utilisant un miroir plan de dimension connue et embarqué sur un robot mobile. L'idée qui sous-tend le concept est de déplacer le miroir embarqué sur le robot mobile de façon à scanner un maximum de parties occultées de l'environnement grâce à la réflexion des faisceaux laser sur la surface du miroir avant l'intervention de l'opérateur humain pour déplacer le TLS une nouvelle fois.

Cette solution est décrite dans la suite de ce chapitre. La section 3.2 présente un état de l'art des travaux qui s'intéressent à l'utilisation des miroirs avec des scanners laser. La section 3.3 présente une synthèse des trois étapes principales de l'approche basée miroir proposée. La première étape de l'approche intitulée "Placement automatique du miroir" est détaillée dans la section 3.4. La seconde étape intitulée "Correction du nuage de points" est présentée dans la section 3.5. La troisième et dernière étape "Sélection de la position suivante du miroir" fait l'objet de la section 3.6. La section 3.7 conclura le chapitre.

3.2 État de l'art

Dans le contexte du SLAM (Simultaneous Localization and Mapping), Yang et al. (Yang & Wang, 2011) ont proposé de combiner l'utilisation d'un scanner laser embarqué sur un robot mobile avec des sonars pour faciliter la détection des surfaces spéculaires et transparentes dans l'environnement à numériser. Dans leur étude, le problème que posent les miroirs et les fenêtres est résolu grâce à la fusion des données des capteurs. (Koch et al., 2016) présentent une approche basée sur des scanners laser multiéchos pour identifier et filtrer les objets réfléchissants. Dans le contexte des scanners laser terrestres (TLS), (Käshammer & Nüchter, 2015) proposent une approche pour détecter automatiquement la présence de miroirs rectangulaires de dimensions connues dans les nuages de points 3D.

A contrario, d'autres travaux s'intéressent à l'utilisation des miroirs pour améliorer les performances des scanners laser. En effet, si les miroirs sont judicieusement placés manuellement ou automatiquement dans des endroits bien choisis au préalable, l'utilisation combinée des miroirs et des scanners laser peut améliorer considérablement le processus de numérisation. Dans leur étude, (Lin et al., 2002) utilisent deux miroirs placés verticalement à proximité d'un visage humain dans le but d'améliorer le suivi des marqueurs pour la reconstruction 3D du visage avec une seule caméra. Par ailleurs, (Fasano et al., 2003) proposent deux scénarios dans lesquels l'utilisation d'un miroir permet de réduire considérablement le temps de numérisation de petits objets à l'aide d'un miroir placé sur une table rotative. Dans les deux cas, une caméra est utilisée pour détecter les marqueurs optiques placés sur les bords du miroir afin d'effectuer une identification précise du plan du miroir.

3.3 Synthèse de l'approche basée miroir

Le planificateur de vues présenté dans le chapitre précédent permet en mode "hors-ligne" de fournir à l'opérateur humain une séquence de positions où placer le

scanner laser terrestre afin d'atteindre un taux de complétude suffisant pour l'ensemble de l'environnement. Après chaque scan, une phase d'analyse de la visibilité permet de générer un ensemble de gaps qui représentent les frontières entre les zones scannées et les zones occultées dans le nuage de points de l'environnement scanner. Par la suite, une stratégie d'exploration est utilisée pour sélectionner la prochaine position du scanner (Next-Best-View) afin de déplacer à nouveau ce dernier dans l'environnement. Le processus itératif est répété jusqu'à ce que la condition d'arrêt soit atteinte.

Afin d'augmenter la couverture de l'espace scanné, il est possible de déplacer le scanner ou d'utiliser d'autres alternatives. L'utilisation d'un miroir plan est une des solutions qui permet cela grâce à la propriété de réflexion des faisceaux laser sur les surfaces spéculaires. Il devient donc possible, en combinant un scanner laser terrestre avec un miroir plan, de numériser des objets placés derrière des gaps sans déplacer à nouveau le TLS.

Dans ce chapitre, nous supposons qu'en plus d'un scanner laser terrestre (TLS), nous disposons d'un système constitué d'un miroir plan supposé parfait et de forme rectangulaire fixée verticalement sur le dessus d'un robot mobile. Pour des raisons de simplicité et sans perte de généralité, l'ensemble du système Robot-Miroir sera par la suite référencé uniquement par le miroir. La figure 3.1 montre une illustration 3D du système TLS/Miroir modélisée sur Blender ainsi que sa représentation en 2D qui sera adoptée par la suite dans ce chapitre. Dans l'espace 2D, le miroir est représenté par un segment 2D noté m , le robot est délimité par une boîte englobante notée BB_R et les objets sont représentés par une coupe horizontale de leurs contours externes à hauteur du TLS.

De manière générale, le processus de numérisation d'un environnement par TLS commence par un placement initial du scanner dans une position définie par l'opérateur humain. Cette position est choisie arbitrairement selon des critères prédéfinis par l'utilisateur lui-même en se basant souvent sur son intuition. Comme défini dans le chapitre précédent et à titre de rappel, un scanner positionné au point L_i produit un nuage de points local noté C_i . Nous supposons que l'ensemble des scans à par-

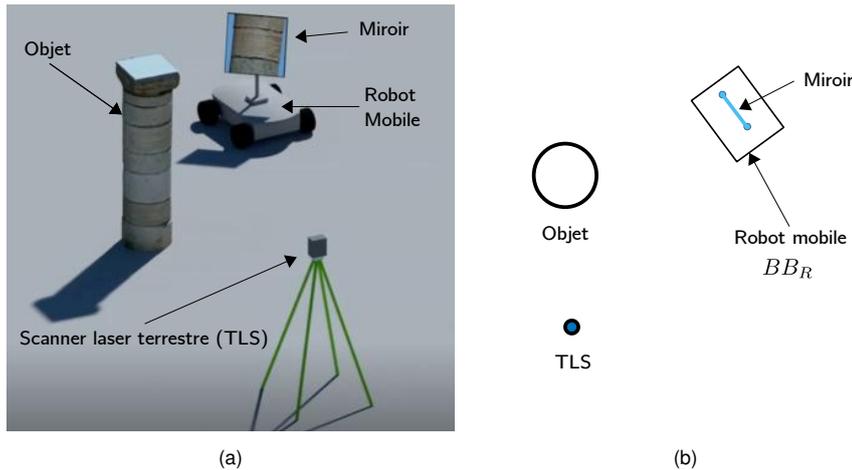


Figure 3.1 – **(a)** Représentation 3D du système TLS/Miroir modélisée sur Blender. **(b)** La représentation en 2D est une vue du dessus de l’environnement à scanner. Dans ce cas, le miroir est un segment noté m délimité par une boîte englobante notée BB_R qui représente le robot.

tir des différentes positions du scanner sont alignés pour produire une représentation globale de l’environnement notée C_0 . Afin d’extraire les zones occultées qui restent à numériser, la Méthode-Projective est utilisée pour détecter l’ensemble des gaps $G_0 = \{^0g_1, ^0g_2, \dots, ^0g_k\}$ dans C_0 où chaque gap 0g_k est défini par ses deux extrémités $(^0\rho_k, ^0\rho_{k+1})$.

L’approche proposée dans ce chapitre intervient à ce niveau et permet de générer pour chaque gap détecté par la Méthode-Projective une position et une orientation optimale du miroir pour scanner l’intérieur du gap sans avoir à déplacer le scanner laser. Ensuite, un planificateur de chemin local est utilisé pour sélectionner parmi les solutions proposées la meilleure position suivante du miroir. Une fois que le miroir est positionné et correctement orienté, un scan dans la zone contenant le miroir est réalisé pour numériser les objets placés derrière le gap associé à la position du miroir précédemment sélectionnée. Ainsi, le nuage de points initial est enrichi itérativement par de nouvelles mesures au fur et à mesure des déplacements du miroir jusqu’à ce que l’ensemble des positions possibles du miroir soit exploité. Si des gaps persistent dans le nuage de points, le planificateur de vues génère une nouvelle position pour

le TLS et l'opérateur humain peut à nouveau intervenir pour déplacer le scanner dans l'environnement.

Une vue d'ensemble de la méthode proposée est illustrée à l'aide de l'organigramme 3.2. On peut distinguer trois nouvelles étapes (en bleu) qui viennent s'ajouter à notre planificateur de vue décrit dans le chapitre précédent (en rouge).

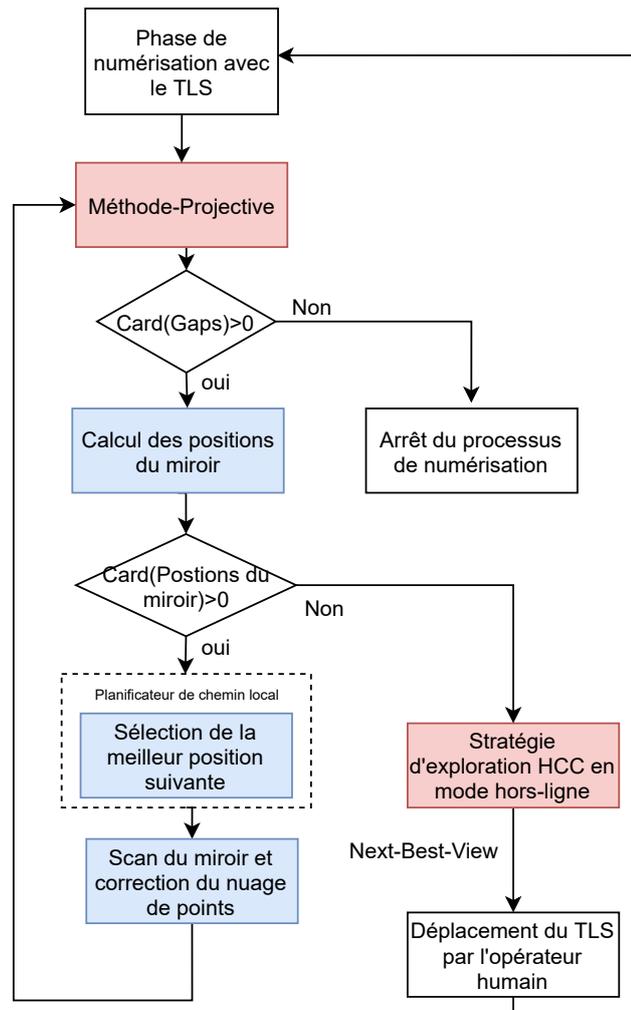


Figure 3.2 – Synthèse de l'approche proposée dans ce chapitre (en rouge) et son interaction avec le planificateur de vue (en bleu) proposé dans le chapitre précédent.

3.4 Placement automatique du miroir

La première étape de l'approche proposée dans ce chapitre consiste à générer de manière automatique pour chaque gap détecté dans l'environnement une position et une orientation optimales du miroir plan afin d'augmenter au maximum la complétude du nuage de points global C_0 sans avoir à déplacer le scanner laser terrestre (TLS).

Étant donné que l'environnement scanné est représenté en 2D, nous supposons que la position d'un miroir noté \mathbf{m} est définie par les coordonnées cartésiennes 2D $(\mathbf{m}_1, \mathbf{m}_2)$ de ses extrémités. Le miroir est également défini par sa largeur $\mathcal{L}_m = \text{dist}(\mathbf{m}_1, \mathbf{m}_2)$ ainsi que par sa normale \mathbf{n}_m qui établit son orientation dans l'espace. De plus, nous supposons que si le scanner laser est positionné au point L_i alors $d_{i \rightleftharpoons m}$ définit la distance orthogonale entre la position du scanner et le plan du miroir \mathbf{m} .

Un miroir plan \mathbf{m} placé dans le champ de vision d'un scanner permet la réflexion totale de l'ensemble des faisceaux laser qui arrivent sur sa surface spéculaire entre ses extrémités $(\mathbf{m}_1, \mathbf{m}_2)$. Le placement optimal d'un miroir plan pour scanner un gap ${}^0g_k = ({}^0\rho_k, {}^0\rho_{k+1})$ consiste à dévier tous les faisceaux réfléchis par le miroir pour scanner l'intérieur du gap. Pour cela, tous les faisceaux laser réfléchis par le miroir doivent passer entre les deux extrémités $({}^0\rho_k, {}^0\rho_{k+1})$.

Pour garantir cela, il suffit de vérifier que les deux faisceaux laser réfléchis par les extrémités du miroir $(\mathbf{m}_1, \mathbf{m}_2)$ arrivent respectivement à scanner les deux extrémités $({}^0\rho_k, {}^0\rho_{k+1})$ du gap 0g_k , comme l'illustre la figure 3.3. L'extension du faisceau laser réfléchi par l'extrémité du miroir \mathbf{m}_1 est une ligne notée l_1 qui passe également par l'une des extrémités du gap ${}^0\rho_k$ ainsi que par l'image virtuelle du scanner laser $L'_i = 2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m$. Il en est de même pour la ligne l_2 qui passe par les trois points \mathbf{m}_2 , ${}^0\rho_{k+1}$ et L'_i . Enfin, l'extension de la ligne qui passe par les deux extrémités du miroir $(\mathbf{m}_1, \mathbf{m}_2)$ permet de définir le plan miroir noté l_3 .

Les lignes l_1, l_2 et l_3 peuvent être définies par les représentations paramétriques

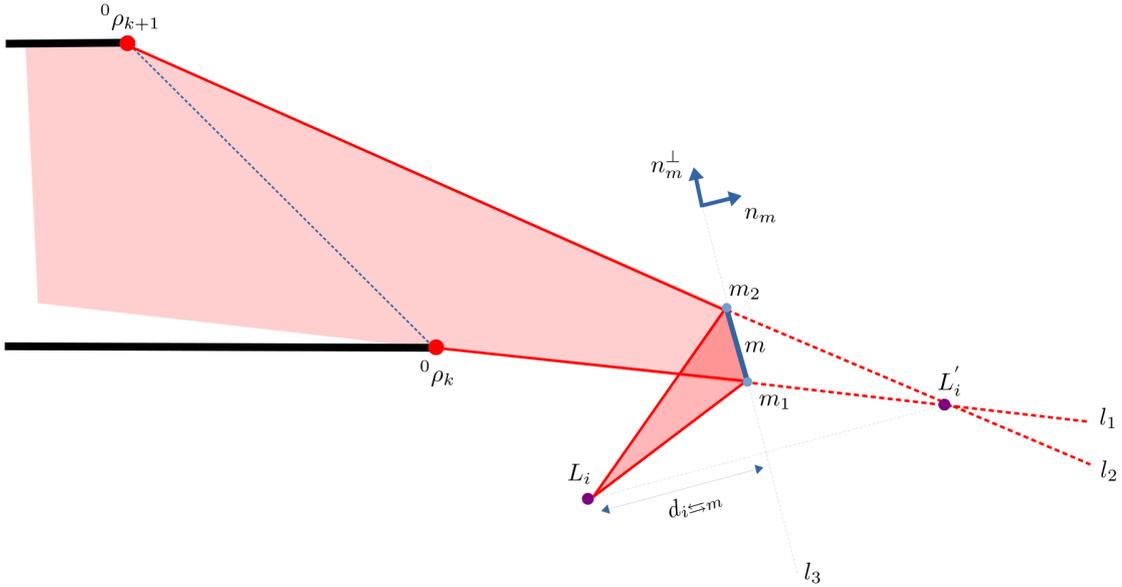


Figure 3.3 – Illustration d'un cas de figure où le placement d'un miroir permet à l'ensemble des faisceaux réfléchis de passer à l'intérieur d'un gap ${}^0g_k = ({}^0\rho_k, {}^0\rho_{k+1})$.

suivantes :

$$\begin{cases} l_1 : 2 \cdot d_{i \Leftarrow m} \cdot \mathbf{n}_m + t_1 \cdot (2 \cdot d_{i \Leftarrow m} \cdot \mathbf{n}_m - {}^0\rho_k) \\ l_2 : 2 \cdot d_{i \Leftarrow m} \cdot \mathbf{n}_m + t_2 \cdot (2 \cdot d_{i \Leftarrow m} \cdot \mathbf{n}_m - {}^0\rho_{k+1}) \\ l_3 : d_{i \Leftarrow m} \cdot \mathbf{n}_m + t_3 \cdot \mathbf{n}_m^\perp \end{cases} \quad (3.1)$$

Où t_1, t_2 et t_3 représentent respectivement les paramètres des représentations des lignes l_1, l_2 et l_3 , \mathbf{n}_m^\perp est le vecteur perpendiculaire à la normale du miroir \mathbf{n}_m .

À partir de cette représentation paramétrique, nous recherchons l'expression des points $(\mathbf{m}_1, \mathbf{m}_2)$. En effet, à partir des extrémités du miroir, nous pouvons déduire sa position ainsi que son orientation définie par sa normale \mathbf{n}_m .

Remarque : \mathbf{m}_1 est l'intersection des lignes l_1 et l_3 et \mathbf{m}_2 est l'intersection des lignes l_2 et l_3 .

Dans le cas de l'extrémité \mathbf{m}_1 , l'intersection des lignes l_1 et l_3 peut-être définie

sous la forme paramétrique suivante :

$$2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m + t_1 \cdot (2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m - {}^0\rho_k) = d_{i \rightleftharpoons m} \cdot \mathbf{n}_m + t_3 \cdot \mathbf{n}_m^\perp$$

En substituant le paramètre t_3 , nous obtenons l'expression de t_1 :

$$t_1 = \frac{d_{i \rightleftharpoons m}}{{}^0\rho_k^T \cdot \mathbf{n}_m - 2 \cdot d_{i \rightleftharpoons m}} \quad (3.2)$$

En substituant la nouvelle expression de t_1 dans l'expression de l_1 (Eq. 3.1), nous obtenons les coordonnées de la première extrémité du miroir \mathbf{m}_1 :

$$\mathbf{m}_1 = 2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m + \frac{d_{i \rightleftharpoons m} \cdot (2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m - {}^0\rho_k)}{{}^0\rho_k^T \cdot \mathbf{n}_m - 2 \cdot d_{i \rightleftharpoons m}} \quad (3.3)$$

Par le même raisonnement, nous pouvons démontrer que l'intersection des lignes l_2 et l_3 permet d'obtenir les coordonnées de la deuxième extrémité du miroir \mathbf{m}_2 :

$$\mathbf{m}_2 = 2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m + \frac{d_{i \rightleftharpoons m} \cdot (2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m - {}^0\rho_{k+1})}{{}^0\rho_{k+1}^T \cdot \mathbf{n}_m - 2 \cdot d_{i \rightleftharpoons m}} \quad (3.4)$$

En pratique, la largeur du miroir plan $\mathcal{L}_m = \text{dist}(m_1, m_2)$ est toujours connue, nous pouvons donc en déduire l'expression suivante :

$$\left\| \frac{2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m - {}^0\rho_{k+1}}{{}^0\rho_{k+1}^T \cdot \mathbf{n}_m - 2 \cdot d_{i \rightleftharpoons m}} - \frac{2 \cdot d_{i \rightleftharpoons m} \cdot \mathbf{n}_m - {}^0\rho_k}{{}^0\rho_k^T \cdot \mathbf{n}_m - 2 \cdot d_{i \rightleftharpoons m}} \right\| = \frac{\mathcal{L}_m}{d_{i \rightleftharpoons m}} \quad (3.5)$$

Connaissant la position des extrémités $({}^0\rho_k, {}^0\rho_{k+1})$ du gap 0g_k à scanner, il suffit de retrouver pour chaque valeur de la distance orthogonale $d_{i \rightleftharpoons m}$ une normale du miroir n_m qui satisfait l'équation (3.5). Il devient donc possible de substituer la valeur de n_m dans l'équation (3.3) pour retrouver la valeur de m_1 et dans l'équation (3.4) pour déterminer la valeur de m_2 .

En faisant varier la valeur de $d_{i \rightleftharpoons m}$ il est possible de générer l'ensemble des positions et orientations possibles du miroir qui garantissent que tous les faisceaux laser réfléchis par le miroir soient réfléchis à l'intérieur du gap 0g_k . Néanmoins, toutes les positions du miroir proposées comme solutions ne sont pas valables (voir Fig 3.4).

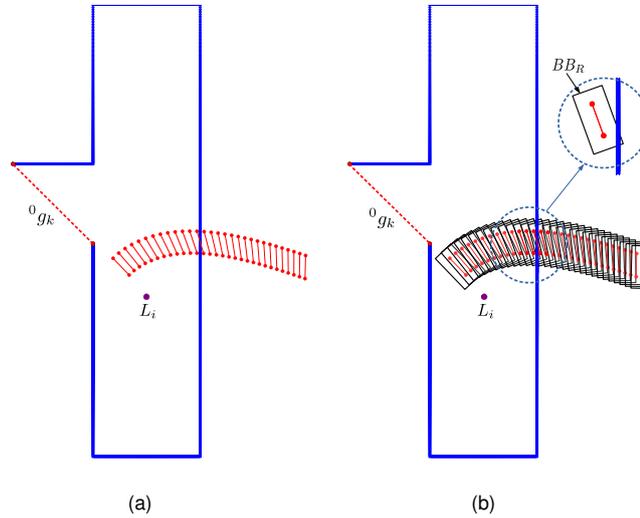


Figure 3.4 – **(a)** L'ensemble des solutions proposées comme positions admissibles du miroir (segments rouges). Certaines positions sont à l'intérieur de l'espace accessible, d'autres ne le sont pas. **(b)** Cas de figure où une collision a lieu entre le robot représenté par la boîte englobante BB_R et un obstacle dans l'environnement.

En effet, certaines positions du miroir sont à l'extérieur de l'environnement visible (scanné). Pour qu'une solution soit valable, elle doit satisfaire la contrainte suivante : la position du miroir doit être à l'intérieur de la zone visible par le scanner laser. De plus, pour éviter des collisions entre le robot et l'environnement, il faut que la boîte englobante notée BB_R qui délimite le robot (et le miroir) soit également à l'intérieur de l'environnement réel ($BB_R \in \Omega_0$).

En intégrant ces contraintes dans l'équation (3.5), on réduit l'ensemble des solutions à des positions valables du miroir comme illustré sur la figure 3.5.

Afin de sélectionner une seule position du miroir parmi toutes les positions valables, nous proposons de sélectionner la position qui permet de maximiser l'angle $\angle(\rho_k, L'_i, \rho_{k+1})$ et par conséquent de maximiser le FOV du scanner laser. Dans l'exemple

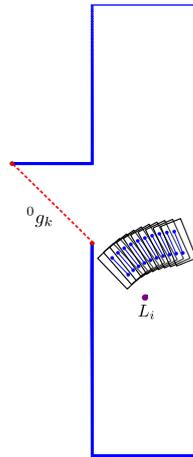


Figure 3.5 – L'ensemble des positions valables pour le miroir. Pour chaque position, la boîte englobante BB_R autour du miroir est à l'intérieur de l'espace visible par le scanner laser.

de la figure 3.5, cette position (Fig. 3.6) se situe à une distance $d_{i \rightarrow m} = 0.71m$ et permet de maximiser le FOV du scanner laser.

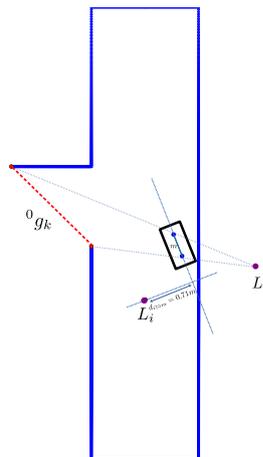


Figure 3.6 – Solution sélectionnée pour être la position suivante du miroir.

3.5 Correction du nuage de points

L'utilisation du système TLS/miroir permet de réduire le déplacement d'un TLS dans le cadre d'un projet de numérisation en augmentant artificiellement le FOV du scanner par l'intermédiaire d'un miroir plan. Cependant, chaque faisceau laser réfléchi par le miroir produit une mesure erronée : le point d'impact est placé derrière le plan réel du miroir. Par exemple sur la figure 3.7 le faisceau laser va parcourir deux fois la distance $d_1 + d_2$ et le scanner au point L_i va considérer que c'est ${}^0\rho'_i$ la valeur scannée au lieu de ${}^0\rho_i$.

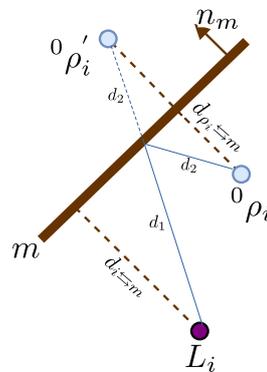


Figure 3.7 – Représentation simplifiée de la problématique engendrée dans le cas d'un faisceau laser réfléchi par un miroir. La mesure retournée au scanner est ${}^0\rho'_i$ au lieu de la mesure correcte ${}^0\rho_i$.

Ce problème est particulièrement difficile à corriger quand la position du miroir n'est pas connue au préalable ; ce qui est le cas notamment dans les travaux de détection des surfaces spéculaires. Dans le cadre notre approche, la position et l'orientation du miroir sont supposées parfaitement connues grâce au système de localisation du robot. Il est donc plus facile de détecter et de corriger les positions des mesures erronées.

3.5.1 Détection des mesures erronées

La connaissance préalable de la position et de l'orientation du miroir permet d'identifier avec précision la position des mesures erronées dans le nuage de points. Après chaque scan du miroir, l'ensemble des mesures erronées est généré derrière le plan miroir dans une zone notée Δ_m délimitée par les points suivants : $\{m_1, m_2, L_i + \rho_{max} \cdot (m_1 \hat{=} L_i), L_i + \rho_{max} \cdot (m_2 \hat{=} L_i)\}$. ρ_{max} est la portée maximale du scanner laser.

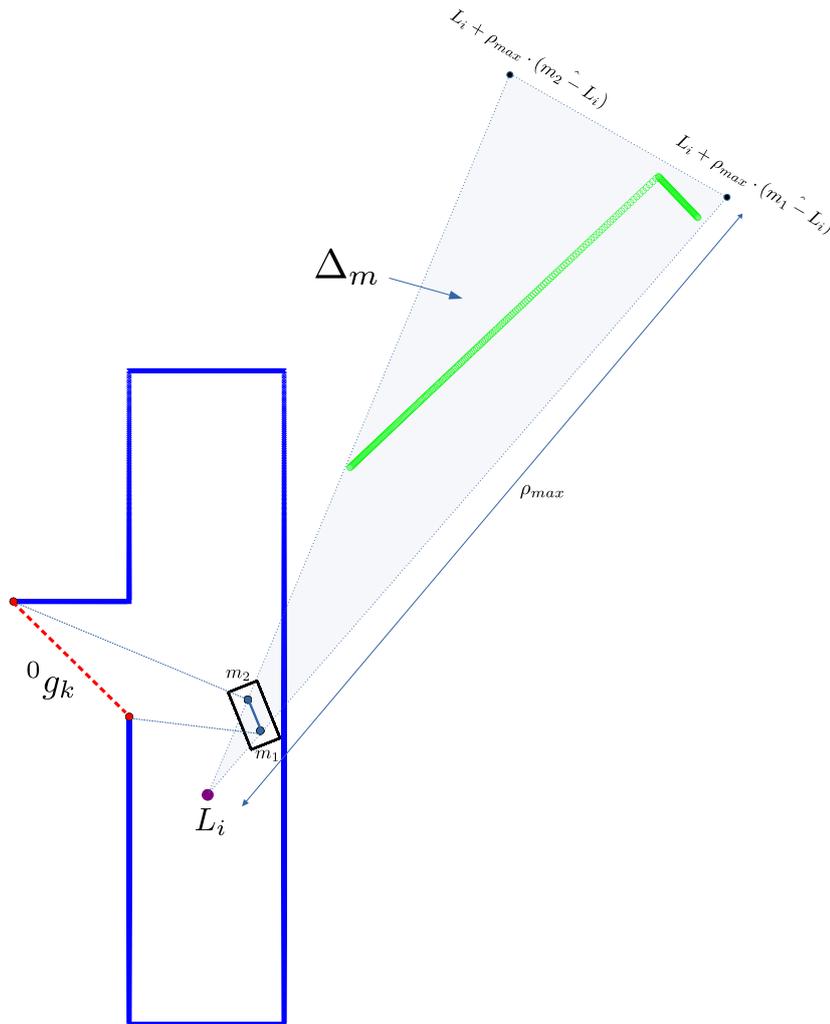


Figure 3.8 – Les mesures incorrectes, ici en vert, sont placées derrière le miroir dans la zone Δ_m

La figure 3.8 illustre un exemple de nuage de points avec des mesures erronées (en vert) placées dans la zone Δ_m . La détection des mesures erronées permet d'extraire leurs coordonnées afin de les corriger par la suite.

3.5.2 Correction des mesures erronées

Afin de corriger les mesures erronées détectées dans la zone Δ_p , nous proposons une méthode basée notamment sur les travaux (Mariottini et al., 2009) et (Marchand & Chaumette, 2017) de reconstruction d'objet par caméra et miroir. Dans notre approche, nous allons remplacer la caméra par un scanner laser.

Pour illustrer le processus de correction, nous reprenons le même schéma de la figure 3.7. Soit ${}^0\rho'_i$ le symétrique du point ${}^0\rho_i$ et $d_{\rho_i \leftrightarrow m}$ la distance orthogonale entre le point ${}^0\rho_i$ (ou ${}^0\rho'_i$) et le miroir m . La relation entre les coordonnées du point ${}^0\rho_i$ de son symétrique ${}^0\rho'_i$ est donnée par :

$${}^0\rho_i = {}^0\rho'_i - 2 \cdot d_{\rho_i \leftrightarrow m} \cdot \mathbf{n}_m \quad (3.6)$$

or la distance $d_{\rho_i \leftrightarrow m}$ est exprimée par :

$$d_{\rho_i \leftrightarrow m} = {}^0\rho'_i \cdot \mathbf{n}_m - d_{i \leftrightarrow m} \quad (3.7)$$

En substituant la valeur de $d_{\rho_i \leftrightarrow m}$ dans l'équation 3.6, nous obtenons :

$${}^0\rho_i = (\mathbf{I} - 2 \cdot \mathbf{n}_m \cdot \mathbf{n}_m^T) \cdot {}^0\rho'_i + 2 \cdot d_{i \leftrightarrow m} \cdot \mathbf{n}_m \quad (3.8)$$

avec \mathbf{I} la matrice identité.

En utilisant la notation matricielle, nous obtenons :

$${}^0\hat{\rho}_i = \begin{pmatrix} \mathbf{I} - 2 \cdot \mathbf{n}_m \cdot \mathbf{n}_m^T & 2 \cdot d_{i \leftrightarrow m} \cdot \mathbf{n}_m \\ 0 & 1 \end{pmatrix} \cdot {}^0\hat{\rho}'_i \quad (3.9)$$

avec ${}^0\hat{\rho}_i$ et ${}^0\hat{\rho}'_i$ les coordonnées homogènes de ${}^0\rho_i$ et ${}^0\rho'_i$.

La figure 3.9 illustre le résultat de la correction des mesures erronées correspondant à la figure 3.8. L'utilisation de la matrice de réflexion 3.9 avec la normale du miroir n_m et sa distance au scanner laser $d_{i\leftrightarrow m}$ permet de corriger les points en vert et de les replacer correctement dans le nuage de points.

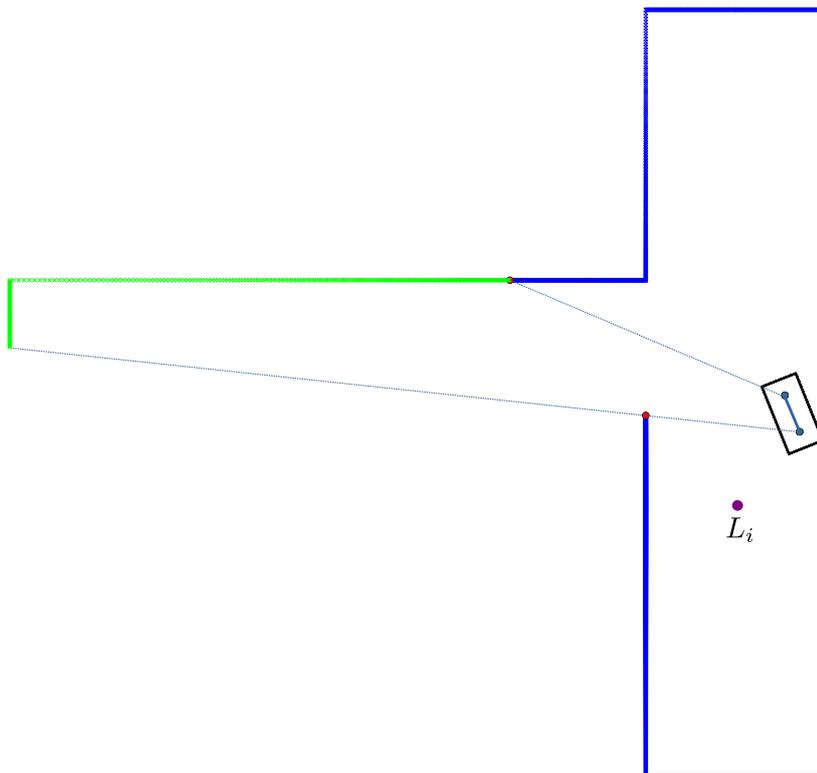


Figure 3.9 – La correction du nuage de points donnée à la figure 3.8 permet de replacer correctement les mesures en vert derrière le gap scanné par le miroir.

3.6 Sélection de la prochaine position du miroir

Dans les deux sections précédentes, nous avons fait le choix de présenter les deux approches proposées dans le cas d'un environnement simple qui présente un

seul gap. En pratique, un environnement complexe présente plusieurs gaps et nécessite donc de générer une position optimale du miroir pour chaque gap détecté.

Dans cette section, nous proposons une stratégie de planification de chemin du robot qui embarque le miroir. Pour ce faire, la position du robot est définie à tout instant par le centre du miroir c_m , supposé connu. L'idée de base de notre algorithme de planification de chemin est similaire à l'algorithme du plus proche voisin (Gutin et al., 2002) et utilise la distance comme seule métrique pour sélectionner la prochaine position du miroir.

Pour cela, l'algorithme de l'entonnoir (Funnel algorithm) (Cui & Shi, 2012) est utilisé pour calculer un chemin optimal sans collision entre la position actuelle du robot c_m et chacune des solutions proposées par notre approche. Par la suite, la solution la plus proche de c_m est sélectionnée pour être la prochaine position du miroir.

La figure 3.10 illustre l'application de notre approche basée sur le système TLS/Miroir dans le cas d'un environnement qui présente plusieurs gaps. Après la première phase de détection des gaps, plusieurs positions du miroir sont proposées pour scanner chacun des gaps dans le nuage de points (Fig 3.10.a).

Ensuite, l'algorithme de planification du chemin choisit la solution la plus proche du scanner laser pour être la position suivante du miroir (Fig 3.10.a). Une fois que le miroir est correctement placé et que le processus de scan du miroir et la correction du nuage de points sont réalisés, une nouvelle phase est lancée (Fig 3.10.b). Le processus itératif est répété jusqu'à ce que l'ensemble des positions possibles du miroir soit exploité. Dans ce cas, l'algorithme permet de couvrir l'ensemble de l'environnement avec le miroir sans avoir à déplacer le TLS (Fig 3.10.c).

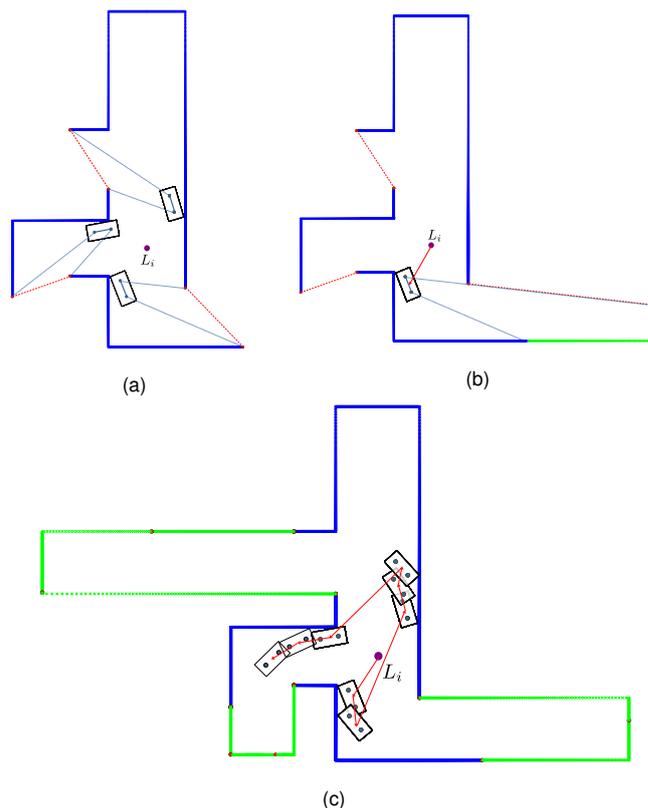


Figure 3.10 – **(a)** Une position du miroir est générée pour chaque gap détecté dans le nuage de points. **(b)** la meilleure position suivante du miroir est sélectionnée par le planificateur de chemin local. Le nuage de points est enrichi par les nouvelles mesures (en vert). **(c)** Le processus itératif est répété pour couvrir l'ensemble de l'environnement.

3.7 Conclusion

Nous avons proposé dans ce chapitre une approche complémentaire qui vient compléter le pipeline (Méthode-Projective+ Cellules conservatives hiérarchiques) dans le cas d'utilisation en mode hors-ligne par une approche basée sur l'utilisation des miroirs plans. Ainsi, nous avons proposé de combiner l'utilisation du Scanner laser terrestre (TLS) avec un miroir plan fixé verticalement sur le dessus d'un robot mobile. D'une part, nous tirons avantage de la réflexion totale des faisceaux lasers sur la surface spéculaire du miroir afin d'augmenter le champ de vision du TLS et de scanner

une grande partie des zones occultées pour le scanner sans avoir à le déplacer dans l'environnement. Le principal avantage de cette approche est que la position du scanner laser reste fixe. Par conséquent, les scans avec les miroirs peuvent être facilement alignés avec le nuage de points initial.

D'autre part, l'utilisation d'un robot mobile permet d'automatiser le déplacement du miroir et ainsi réduire considérablement l'effort fourni par l'opérateur humain. En effet, le miroir permet un gain important en taux de complétude du nuage de points initial sans avoir à bouger le TLS de son emplacement initial.

Néanmoins, il est nécessaire de rappeler que l'utilisation de cette approche basée miroir est limitée au cas "hors-ligne", car son utilisation et sa conception a été réfléchie pour s'adapter à une utilisation avec Scanner laser terrestre. En effet, l'utilisation d'un scanner mobile avec un planificateur de vue en mode "en ligne" ne nécessite pas d'intervention supplémentaire de l'opérateur humain, car la numérisation de l'environnement est totalement automatisée.

Dans le chapitre suivant "Résultats expérimentaux", la méthode proposée dans ce chapitre sera dans un premier temps simulée dans un cas d'utilisation simple et complexe puis comparée avec notre planificateur de vue présenté dans le chapitre 2 en mode "Hors-ligne".

4.1 Introduction

Dans ce chapitre nous présentons quelques résultats expérimentaux sur des données synthétiques et réelles pour valider la capacité de notre approche à résoudre le problème de la planification des vues dans des environnements simples et complexes ainsi que la faisabilité de son implémentation sur des systèmes réels avec l'utilisation d'un robot mobile et des données issues d'un TLS.

Nous avons effectué dans un premier temps des tests sur des données synthétiques générées par un simulateur que nous avons développé à cette fin. Ce simulateur permet de créer des environnements 2D, de sélectionner les positions souhaitées du scanner laser et de générer les nuages de points 2D locaux et globaux.

Il devient ainsi possible de détailler et analyser le fonctionnement du planificateur de vue dans les trois modes d'utilisation suivants :

- **Mode 1** : Utilisation du planificateur de vue présenté dans le chapitre 2 en mode hors-ligne avec un scanner laser terrestre
- **Mode 2** : Utilisation du planificateur de vue présenté dans le chapitre 3 avec l'approche basée miroir.
- **Mode 3** : Appelé mode "en ligne" ou "temps réel" et il représente l'utilisation du planificateur de vues présenté dans le chapitre 2 avec un scanner laser embarqué sur un robot mobile.

En outre, le simulateur nous permet d'évaluer les performances de notre méthode en réalisant plusieurs études comparatives :

- Une comparaison entre les **Mode 1** et les performances d'un opérateur humain expérimenté dans le domaine de la numérisation d'environnements complexes.
- Une comparaison entre le planificateur de vues en **Mode 3** et les quatre approches de la littérature suivantes : (Nasir et al., 2015; Yamauchi, 1997; González-Banos & Latombe, 2002; Mobarhani et al., 2011).

Par la suite, nous considérons le cas de l'implémentation sur une architecture réelle. Pour ce faire, un robot mobile équipé d'un scanner 2D est utilisé en Mode 3 pour explorer et réaliser un scan complet d'un environnement intérieur.

La section 4.2 détaille le fonctionnement des trois modes du planificateur de vue sur le simulateur dans le cas d'environnements simples et complexes. La section 4.3 présente deux études comparatives dans le cas des modes 1 et 3. Par la suite, on considère dans la section 4.4 l'implémentation du mode 3 sur un système réel. La section 4.5 conclut le chapitre.

4.2 Simulations

Les résultats obtenus avec notre simulateur dans le cas d'un environnement simple sont présentés dans le cadre de trois scénarios (Mode 1, Mode 2 et Mode 3). Le premier scénario simulé décrit le fonctionnement de notre approche lorsque le capteur est un scanner laser terrestre (TLS) déplacé par un opérateur humain. Dans le second scénario, nous présentons l'utilisation d'un scénario qui décrit le fonctionnement du système TLS/Miroir. Finalement, la configuration d'un capteur embarqué sur un robot mobile en Mode 3 est présentée dans le dernier scénario.

4.2.1 Mode 1 : Simulation avec un scanner laser terrestre (TLS)

Le déplacement du scanner laser terrestre est effectué par un opérateur humain sur la base de ses observations et de sa connaissance préalable de l'environnement. L'objectif principal de cette simulation est de démontrer l'intérêt d'utiliser le planificateur de vues comme solution d'aide à la décision pour choisir la prochaine position de la

numérisation.

Nous partons de l'hypothèse que l'opérateur humain est équipé d'un TLS et qu'il souhaite numériser un environnement dont une vue du dessus est fournie dans la figure 4.1a. Pour cela, l'opérateur fait le choix de déplacer le scanner sur les quatre positions notées L_1 , L_2 , L_3 et L_4 sur la figure 4.1a. Il décide ensuite d'évaluer la complétude du nuage de points pour prendre une décision. À cette fin, la Méthode-Projective est utilisée pour analyser la visibilité dans le nuage de points global et de visualiser sous forme des fragments représentant les frontières entre les zones préalablement scannées et les zones occultées (Fig. 4.1b).

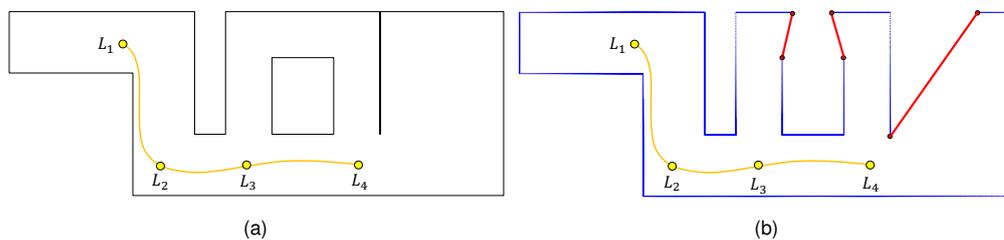


Figure 4.1 – **(a)** Environnement réel (lignes noires) avec les quatre positions du scanner (points jaunes). **(b)** Le nuage de points global C_0 avec les gaps associés aux frontières entre la zone scannée et les zones occultées.

À ce stade, la stratégie d'exploration (HCC) va permettre d'aller plus loin en proposant directement à l'opérateur une prochaine position optimale du scanner (Next-Best-View). Cette position est notée L_5 sur la Fig. 4.2. Une fois le TLS déplacé, le planificateur de vues détecte qu'il reste une zone occultée dans le nuage de points et propose comme position suivante L_6 qui sera la dernière.

4.2.2 Mode 2 : Simulation du système TLS/Miroir

Dans ce scénario, nous considérons le même environnement que celui de la section précédente (Figure 4.1.a) pour valider le planificateur de vues en Mode 2. L'opérateur dispose cette fois-ci d'un TLS et d'un robot mobile équipé d'un miroir plan.

L'objectif est de numériser l'ensemble de l'environnement avec le moins de déplacements possibles. Il commence par placer le TLS au point L_1 et lance une phase de

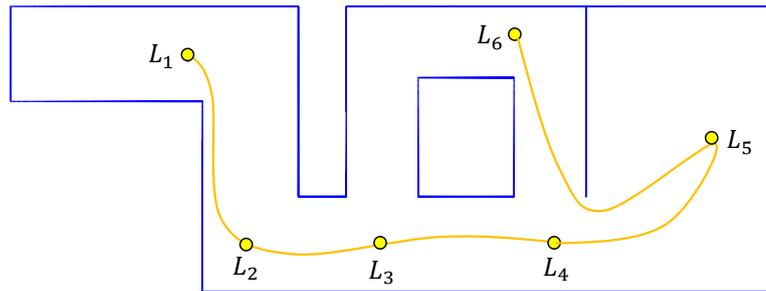


Figure 4.2 – L’opérateur humain procède à une numérisation partielle en déplaçant le scanner quatre fois aux points L_1 , L_2 , L_3 et L_4 . Le planificateur de vues calcule ensuite automatiquement les positions suivantes L_5 puis L_6 pour scanner les gaps restants.

numérisation. Les données acquises sont ensuite traitées dans l’analyse de visibilité avec la Méthode-Projective pour détecter le gap g_1 . Ensuite, le planificateur de vues cherche une position valable du miroir associée au gap g_1 . Dans ce cas, une solution est trouvée dans le champ de vue du scanner laser (Figure 4.3.a) et un scan est effectué en ciblant le miroir pour enrichir le nuage de points avec les nouvelles mesures illustrées sur la figure 4.3.b en vert. Le nouveau nuage de points permet de voir d’autres régions de l’environnement et de détecter trois nouveaux gaps g_2 , g_3 et g_4 .

Une nouvelle position du miroir est détectée pour le gap g_2 et le même processus est répété (Figure 4.3.c). Dans le cas des deux autres gaps g_3 et g_4 , il est impossible de trouver une position valable accessible au TLS à partir du point L_1 .

À ce stade du processus de numérisation, il reste plusieurs gaps dans le nuage de points, mais le planificateur de vues n’arrive plus à trouver de positions valables pour le miroir, il propose donc à l’opérateur humain une nouvelle position L_2 du TLS. L’opérateur intervient dans le processus pour déplacer le scanner au point L_2 et lancer une nouvelle phase de numérisation.

Ce processus itératif est répété jusqu’à ce que l’ensemble de l’environnement soit numérisé (Figure 4.3.d). Le système TLS/miroir permet à l’opérateur humain de scanner l’ensemble de l’environnement avec seulement trois déplacements du TLS, réduisant ainsi considérablement le nombre de positions du TLS et par conséquent l’effort fourni par l’opérateur humain durant toute la phase de numérisation. À titre d’exemple, la

numérisation de l'environnement illustrée sur la figure 4.4 n'a nécessité que deux déplacements du TLS pour obtenir un nuage de points complet.

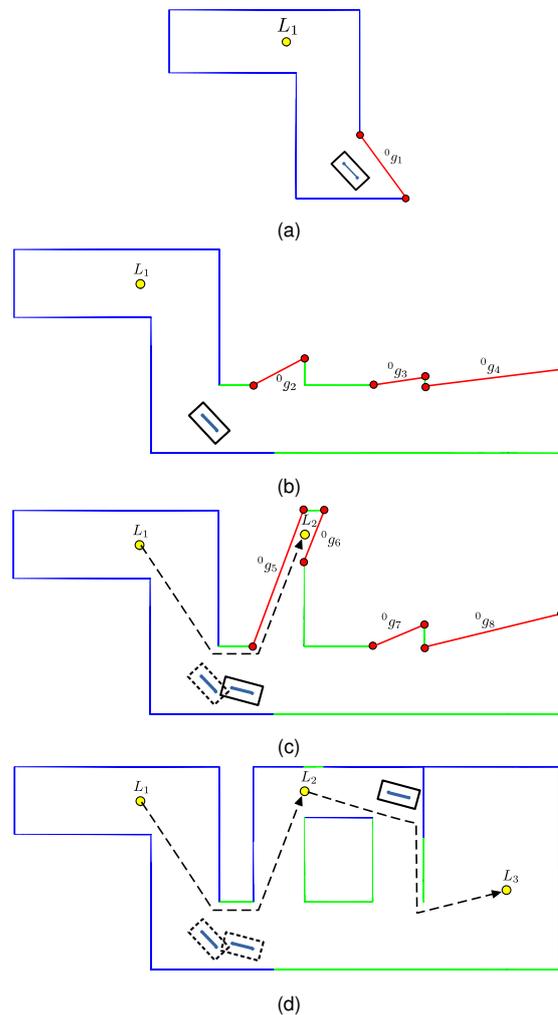


Figure 4.3 – **(a)** Détection d'un gap avec une position valable du miroir pour le scanner. **(b)** Le scan du miroir révèle trois nouveaux gaps g_2 , g_3 et g_4 . **(c)** Une nouvelle position du scanner L_2 est générée après Le scan du gap g_2 avec le miroir. **(d)** Le nuage de points final avec les trois positions du scanner L_1 , L_2 et L_3 ainsi que les trois positions du miroir.

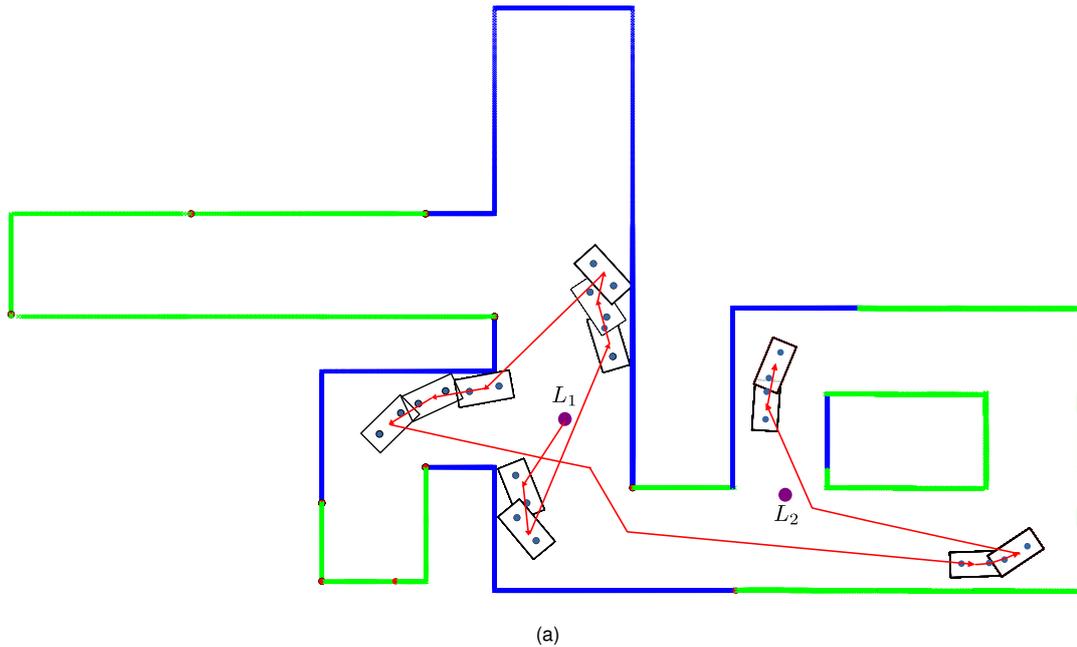


Figure 4.4 – Utilisation du planificateur de vues en Mode 2 dans le cas d'un environnement complexe. Les points bleus sont scannés directement avec le TLS alors que les points verts sont scannés avec miroir.

4.2.3 Mode 3 : Simulation avec un robot mobile équipé d'un scanner laser

La simulation suivante décrit le scénario dans lequel un robot mobile est équipé d'un scanner laser en vue d'explorer automatiquement son environnement. À la différence d'un humain, le robot est dépourvu d'une vue globale de l'environnement qui lui permettrait de planifier une suite positions. De ce fait, le planificateur de vues se révèle nécessaire dès le premier scan et ce, jusqu'à la fin de la numérisation.

La figure 4.5 montre le déroulement du processus où L_1 désigne la position initiale du robot. Depuis cette position, la Méthode-Projective est utilisée pour détecter les zones occultées qui restent à explorer. Le résultat obtenu sur la figure 4.5a fait apparaître deux zones inexplorées désignées par R_1 et R_2 situées derrière trois fragments, ici en rouge. Ensuite, le planificateur de vues génère trois candidats (c_1, c_2, c_3) associés aux trois fragments précédemment détectés, puis sélectionne c_1 comme Next-Best-View (NBV).

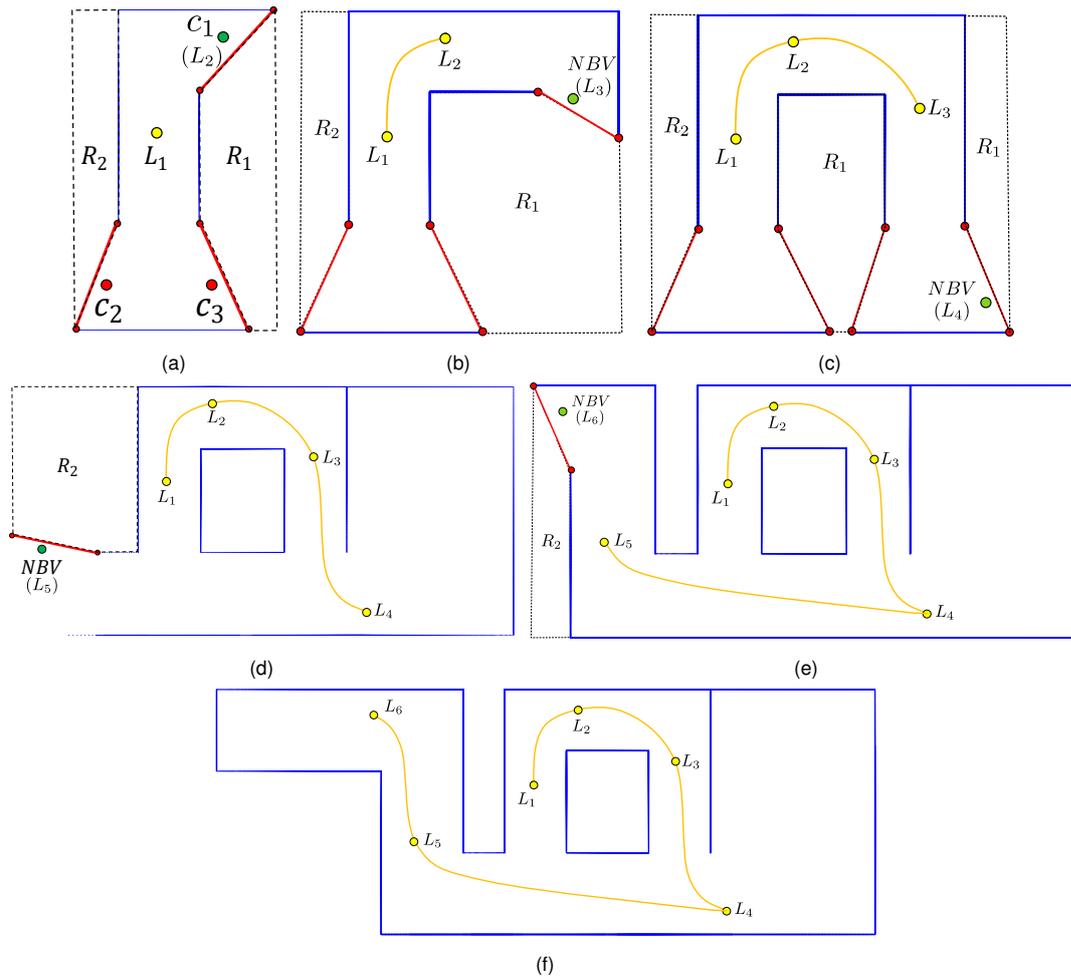


Figure 4.5 – **(a)** Le processus de détection des gaps révèle deux zones occultées à proximité de la position initiale du robot L_1 . **(b)-(e)** Le robot explore l'ensemble de la région R_1 avant de passer à la région R_2 **(f)** La trajectoire finale du robot obtenue au moyen de notre stratégie d'exploration.

Les figures 4.5(b-e) représentent toutes les étapes intermédiaires du processus d'exploration durant lequel le robot est resté à l'intérieur de la région R_1 . Ceci s'explique par le fait que le planificateur de vues utilise une contrainte de distance dans le processus de sélection de la NBV et qui contraint le robot à explorer complètement une région avant de se diriger vers une autre. Ainsi, la figure 4.5f montre que la trajectoire finale du robot ne comporte pas de mouvements de va-et-vient inutiles entre les régions R_1 et R_2 au cours de l'exploration.

Ultérieurement, notre planificateur de vues a également été évalué dans des environnements plus complexes, comme le montre la figure 4.6. Dans ce type d'environnement, il devient relativement compliqué, même pour un opérateur humain expérimenté de déterminer facilement un nombre optimal de positions du scanner laser pour couvrir la totalité de l'environnement. La simulation montre que notre approche permet de générer une séquence de positions de manière à ce qu'il n'y ait plus de fragments dans le nuage de points global (en bleu dans la figure 4.6). De plus, la trajectoire du robot montre que les contraintes sélectionnées dans notre approche, à savoir la distance et le gain d'informations, permettent au robot d'éviter des mouvements de va-et-vient inutiles dans l'environnement.

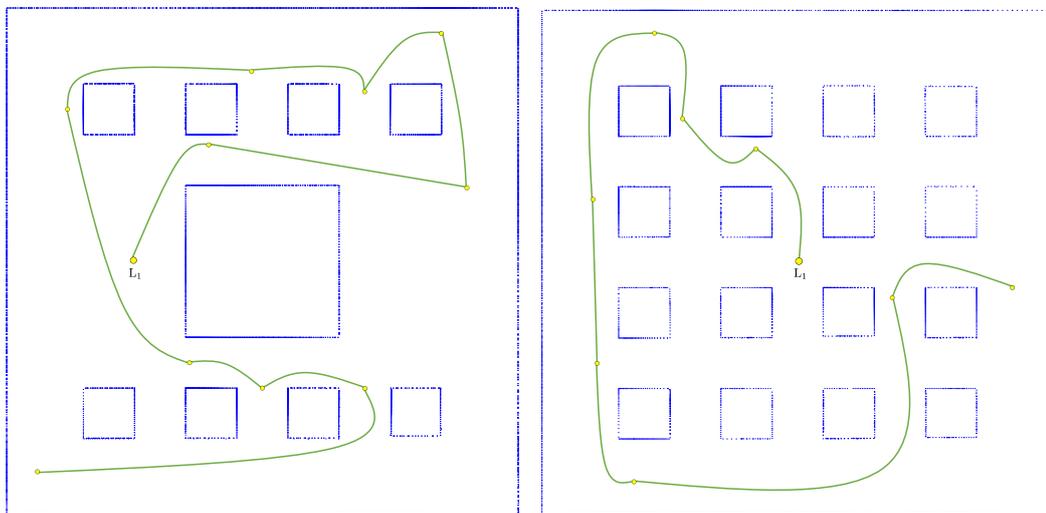


Figure 4.6 – Les chemins générés par le planificateur de vues en Mode 3 dans le cas d'environnements plus complexes.

4.3 Études comparatives

Dans l'optique d'évaluer les performances du planificateur de vues proposé, nous avons commencé par comparer les performances de l'approche en Mode 1 avec les décisions d'un opérateur humain expérimenté. Une seconde étude comparative est dédiée à la comparaison entre le Mode 3 et des méthodes de la littérature.

4.3.1 Mode 1 : Analyses des performances en mode hors-ligne

Les opérateurs humains sont particulièrement efficaces quand il s'agit de planifier la numérisation d'un environnement simple. Ces emplacements sont déterminés empiriquement en fonction du savoir-faire de l'opérateur et des caractéristiques du site à numériser. Cependant, cette tâche s'avère ardue dans des environnements larges et complexes. Pour assister l'opérateur humain, des algorithmes de planification de vues sont employés pour calculer les positions optimales et réduire au maximum le nombre d'emplacements du scanner.

Afin d'évaluer les performances de notre algorithme de planification de vues, nous nous sommes basés sur les données de numérisation réalisée par un opérateur humain lors d'une campagne de numérisation dans un environnement complexe.



Figure 4.7 – **(a)** L'esplanade de Hassan à Rabat (Maroc). **(b)** Nuage de points 3D de cette esplanade après 17 scans.

En juin 2015, dans le cadre du projet ATHAR3D (Achakir et al., 2017), une numérisation de l'esplanade de Hassan à Rabat (Fig. 4.7a) a été réalisée par lasergrammétrie (scanner Leica C10). L'opérateur a effectué 17 scans sur l'ensemble de l'environnement qui dépasse les 2000 m^2 afin de couvrir l'ensemble des 500 colonnes et les restes de l'enceinte entourant l'esplanade de la mosquée. Le nuage de points 3D final est présenté sur la figure 4.7b.

Afin d'effectuer une comparaison avec les résultats de l'opérateur humain, nous avons extrait une coupe 2D de ce nuage de points (Fig. 4.8a) pour recréer sur notre

simulateur un environnement similaire à une vue du dessus de l'esplanade (Fig. 4.8b).

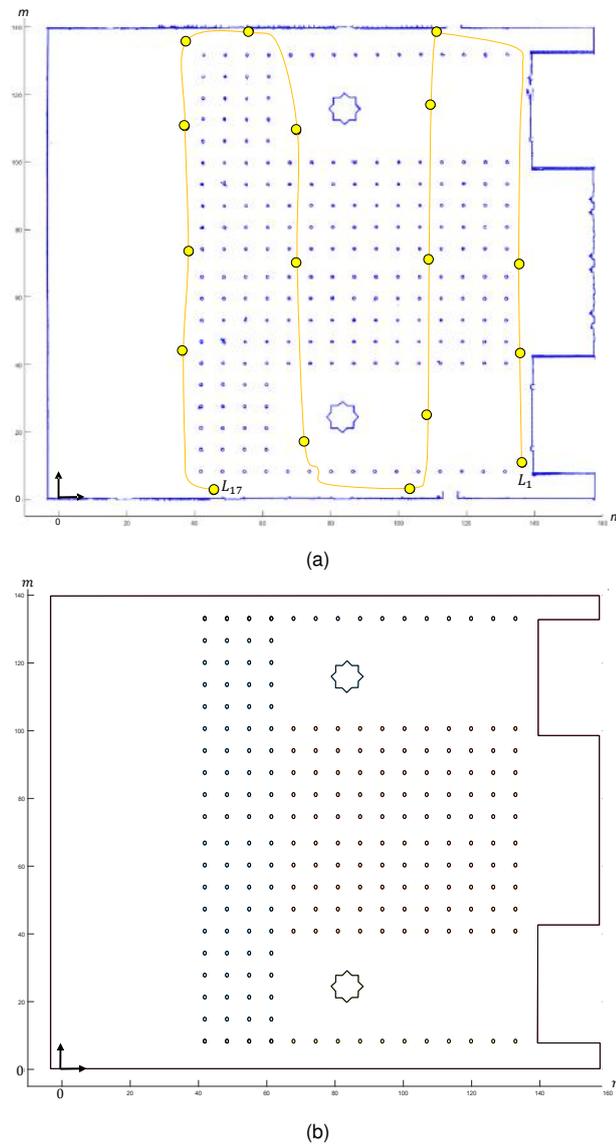


Figure 4.8 – **(a)** Extrait 2D du nuage de points (bleu) et les 17 positions du scanner utilisées pour couvrir le centre de l'esplanade. **(b)** L'environnement de l'esplanade récréé avec le simulateur.

Par la suite, le simulateur a été configuré pour exécuter le planificateur de vues à partir de la même position initiale choisie par l'opérateur humain. La séquence de positions planifiées par l'algorithme est fournie dans la figure la figure 4.9.

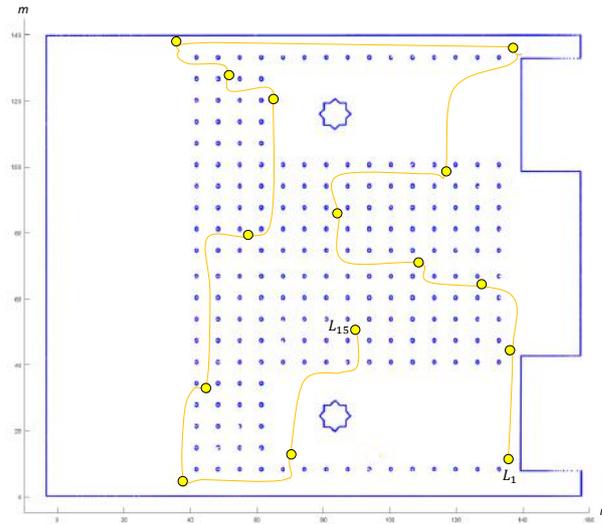


Figure 4.9 – Résultats du calcul des candidats avec notre planificateur de vues dans l’environnement simulé du centre de l’esplanade de Hassan.

Afin de comparer les performances de notre approche et celle de l’opérateur humain, nous avons eu recours au taux de couverture de la scène comme critère de comparaison. Le simulateur nous permet d’estimer ce taux de couverture à l’intérieur de la région A_0 (Fig. 4.10a) après chaque déplacement du scanner. Ce taux de couverture est défini de la manière suivante :

$$coverage(\%) = \frac{Area(\Omega_0)}{Area(A_0)} \cdot 100 \quad (4.1)$$

Où Ω_0 (équation 2.10) est la fusion de l’ensemble des polygones de visibilité locaux qui représente l’espace scanné.

Les résultats de la figure 4.10b montrent que notre approche génère 15 positions assurant un taux de couverture proche de 99,2%, ce qui est similaire aux résultats obtenus par l’opérateur humain (99,08% après 15 déplacements du scanner et 99,26% après 17 déplacements du scanner). Dans le cas de notre planificateur de vues, le processus de numérisation est interrompu lorsque la longueur de tous les fragments restants dans l’environnement est inférieure à un seuil fixé à 0,5m dans cet exemple. Nous constatons également que notre méthode permet d’atteindre un taux de cou-

verture plus élevé (94.5% avec 9 positions) avec un nombre de positions du scanner inférieur à celui de l'opérateur humain (95% avec 12 positions).

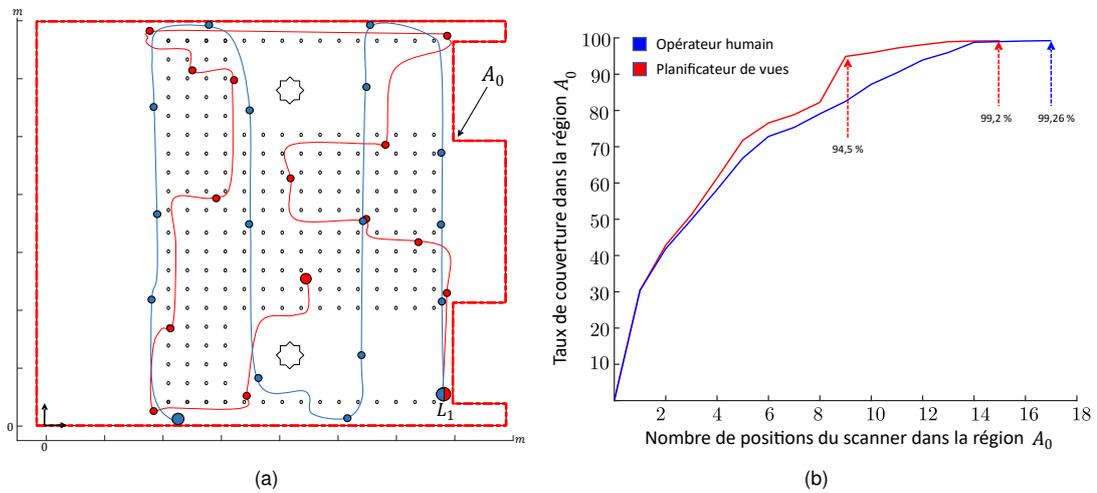


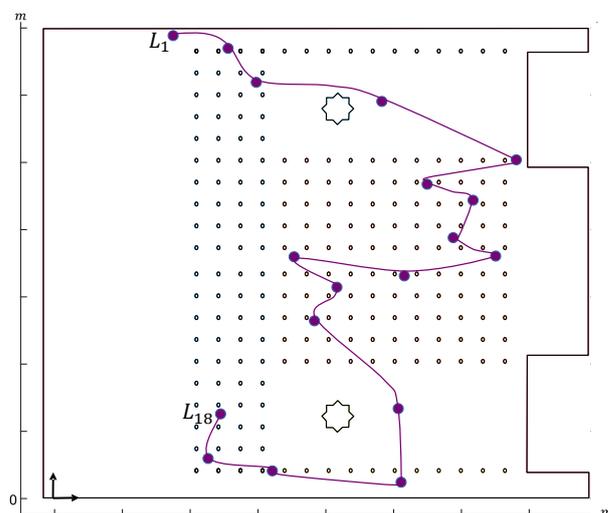
Figure 4.10 – **(a)** Comparaison entre les positions générées avec notre approche (rouge) et le choix de l'opérateur humain (bleu) dans la région A_0 qui délimite le centre de l'esplanade. **(b)** Le taux de couverture à l'intérieur de la région A_0 après chaque déplacement du scanner.

Pour atteindre le taux de couverture maximal (100%) avec notre approche il a fallu baisser le seuil à $0,01m$. Le nombre de positions proposées par notre approche suite à ce seuil est de 27 positions.

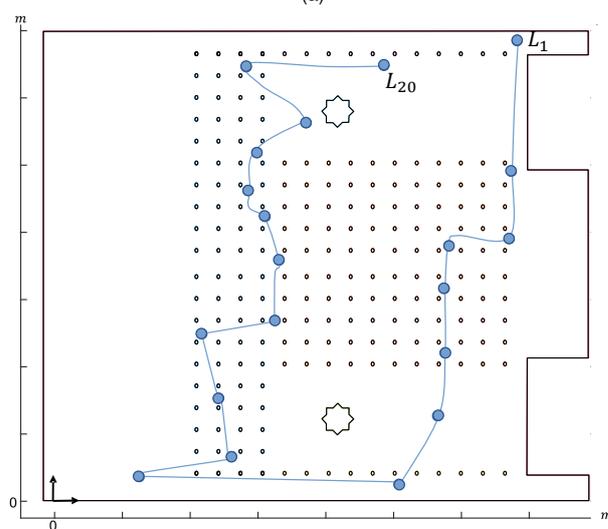
De plus, nous avons modifié délibérément le choix de la position L_1 pour évaluer son impact sur les performances du planificateur de vues. Pour ce faire, nous avons choisi deux nouvelles positions initiales L_1 et nous avons utilisé notre planificateur de vues dans l'environnement de simulation de l'esplanade de Hassan (figure 4.8.b). La figure 4.11 montre que le choix de la position L_1 fait varier le nombre de positions finales du scanner laser produites par le planificateur de vues. Le nombre final de positions pour assurer la couverture maximale de l'environnement scanné peut varier de 17 positions comme c'est le cas de la figure 4.10.a à 18 ou 20 positions pour d'autres positions L_1 comme c'est le cas sur la figure 4.11.

Il est également nécessaire de préciser que de manière générale pour déterminer

la position initiale optimale dans un environnement quelconque, il faut évaluer toutes les positions initiales possibles et donc disposer d'un modèle préalable complet de l'environnement ce qui rentre dans le cadre des planificateurs de vues avec un modèle préalable (Model-Based views planners).



(a)



(b)

Figure 4.11 – Résultats du calcul des positions du scanner laser avec notre planificateur de vues pour différentes positions initiales L_1 . Le planificateur de vues permet d'assurer la couverture de l'environnement après 18 (a) ou 20 positions (b) du scanner laser.

4.3.2 Mode 3 : Analyses des performances en mode temps réel

La planification automatique des vues a été étudiée dans la littérature et plusieurs approches employant différentes stratégies pour obtenir une couverture complète de l'environnement à explorer, ont été proposées. Très fréquemment, les stratégies d'exploration visent à déterminer le chemin le plus court pour explorer l'ensemble de l'environnement en un minimum de temps. Quatre approches ont été retenues pour la comparaison avec notre approche : Gap navigation tree (GNT) (Nasir et al., 2015), Frontier-Based Approach for Autonomous Exploration (Yamauchi) (Yamauchi, 1997), Histogram based frontiers (HBF) (Mobarhani et al., 2011) et une approche volumétrique (Gonzalez) (González-Banos & Latombe, 2002). Le tableau (Tableau 4.1) résume chacune des méthodes sélectionnées :

Les méthodes sélectionnées ont été simulées dans le même environnement que celui illustré dans la Fig. 4.12. Toutes les méthodes sont initialisées avec la même position du scanner au point L_1 , et à chaque nouvelle position, le temps de calcul ainsi que la longueur du chemin cumulée ont été enregistrés.

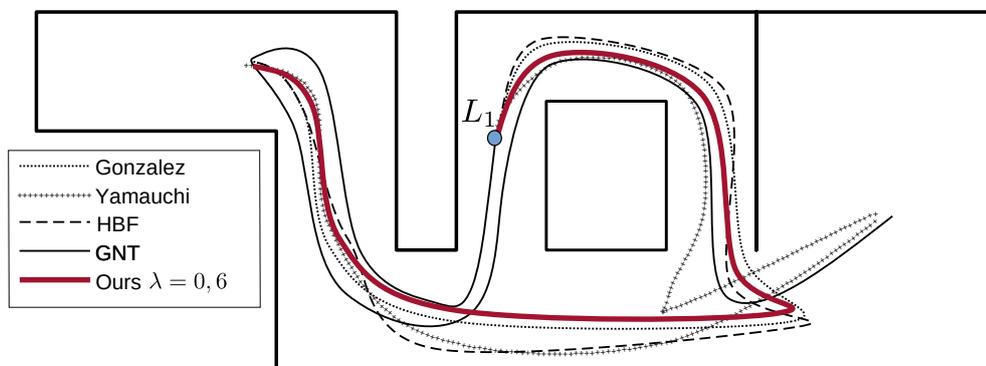


Figure 4.12 – Trajectoires finales générées avec chacun des planificateurs de vues sélectionnés.

Comme le montre la figure 4.13a, Yamauchi (Yamauchi, 1997) et HBF (Mobarhani et al., 2011) sont très chronophages en raison du recours au lancer de rayons

Planificateur de vues	Représentation de l'environnement	Stratégie d'exploration
Yamauchi (Yamauchi, 1997)	Grille d'occupation 2D	Le centre de la frontière inexplorée la plus proche du robot est sélectionné comme NBV.
HBF (Mobarhani et al., 2011)	Grille d'occupation 2D	Une fonction d'utilité est utilisée pour évaluer le score de chaque frontière dans la carte de la grille. Le centre de la frontière ayant le score le plus élevé est sélectionné pour être la NBV.
Gonzalez (González-Banos & Latombe, 2002)	Nuage de points/Gaps	Un ensemble de candidats est généré aléatoirement à proximité de chaque écart. Le candidat ayant le score le plus élevé est sélectionné comme NBV.
GNT (Nasir et al., 2015)	Nuage de points/GNT	l'arbre GNT est utilisé pour classer les gaps par ordre de détection. Le robot explore l'arbre depuis les nœuds ancêtres (premiers gaps détectés) jusqu'aux nœuds descendants (derniers gaps détectés).

TABLE 4.1 – Synthèse des quatre approches sélectionnées pour la comparaison avec notre approche.

avec la représentation de la carte de l'environnement en grille d'occupation. Gonzalez (González-Banos & Latombe, 2002) a proposé une stratégie qui évite le lancer de rayons avec une représentation de l'environnement basée sur le nuage de points, mais l'utilisation d'un grand nombre de candidats pour déterminer la NBV du scanner entraîne une forte augmentation du temps de calcul. En revanche, la GNT produit des résultats très intéressants en termes de temps de calcul, mais la simplicité de cette stratégie d'exploration se traduit par un long chemin cumulatif comme le montre la Fig. 4.13b.

En revanche, notre approche (Ours $\lambda = 0.6$) ne nécessite pas de lancer de rayons et réduit le nombre de candidats grâce aux "cellules conservatrices". De plus, l'utilisa-

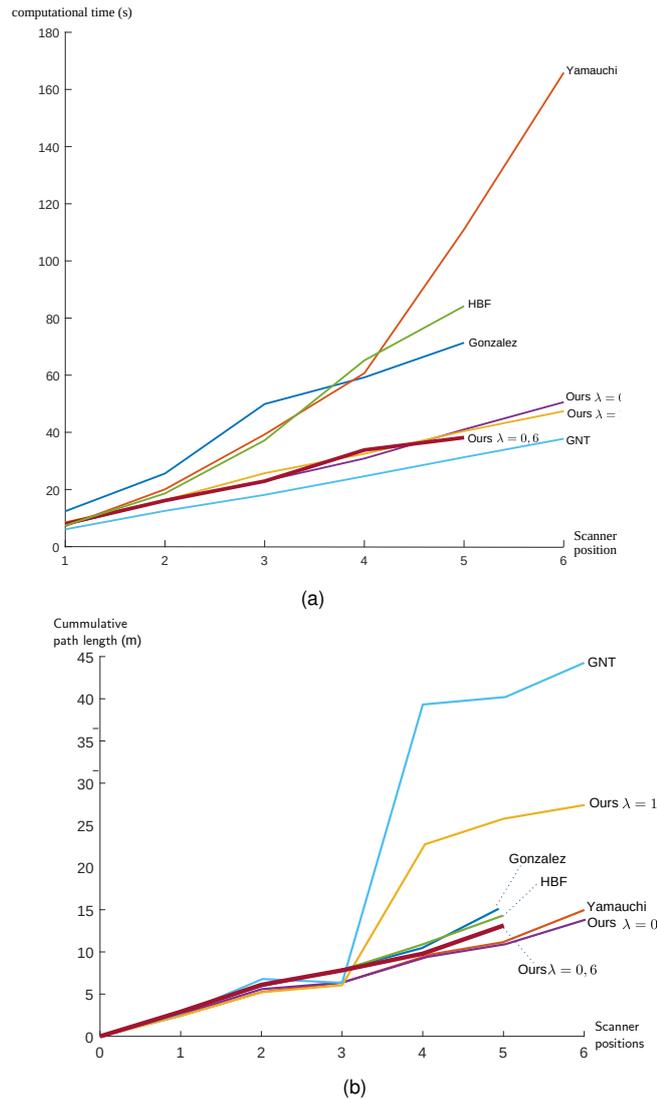


Figure 4.13 – **(a)** Comparaison des performances en fonction du temps de calcul. **(b)** Comparaison des performances en fonction de la longueur du chemin cumulé.

tion d'une fonction d'utilité pour évaluer le score des candidats nous permet d'éviter un long chemin cumulatif avec seulement cinq positions du scanner. Une stratégie basée uniquement sur la distance comme c'est le cas pour "Yamauchi, Ours $\lambda = 0$ " ou uniquement sur le gain d'information attendu comme "Ours $\lambda = 1$ " génère six positions pour couvrir tout l'environnement.

4.4 Expérimentation en temps réel avec un robot mobile (Mode 2)

Dans les simulations précédentes, nous avons validé le fonctionnement de notre approche dans les trois modes de fonctionnement. Dans cette expérimentation, nous cherchons à valider l'implémentation et le fonctionnement de notre approche sur une architecture réelle, ainsi que sa capacité à résoudre le problème de la planification des vues avec des données réelles issues d'un scanner laser embarqué sur un robot mobile. Pour ce faire, nous avons choisi de nous orienter vers un schéma expérimental divisé en deux modules principaux synthétisés dans l'organigramme de la figure 4.14.

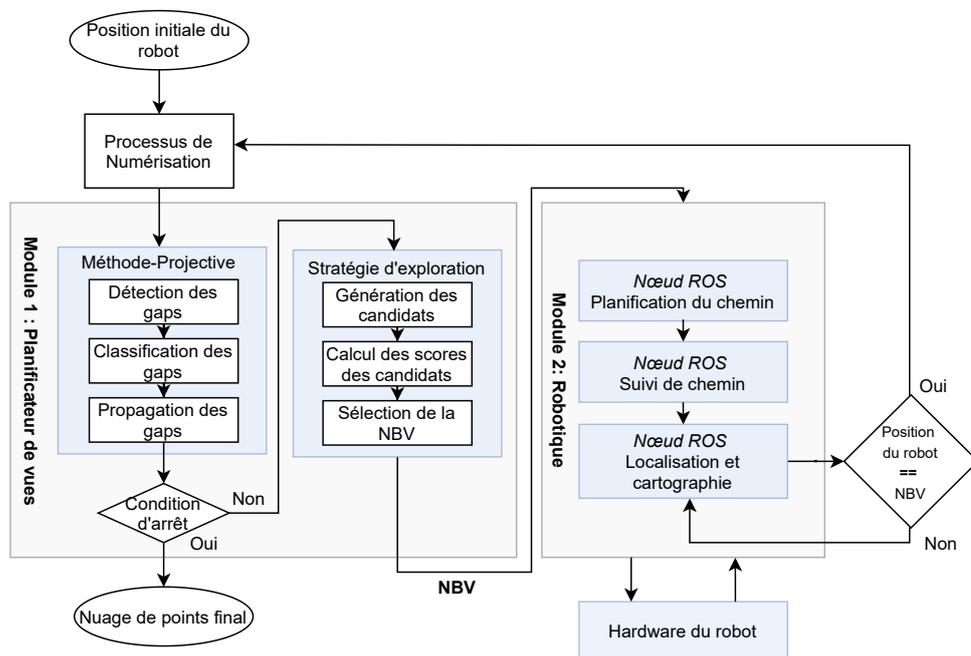


Figure 4.14 – Schéma expérimental décrivant l'interaction entre le module "planificateur de vues" et le module "Robotique".

Le premier module représente notre planificateur de vues. Il permet principalement de détecter et de classer les gaps, d'identifier les zones occultées dans le nuage de points de l'environnement et de calculer la NBV. Le second module "Robotique" est principalement composé de nœuds ROS (Robot Operating System). Il permet la localisation du robot dans son environnement, la planification et le suivi de chemin.

Cette expérience a été réalisée avec un robot mobile Pioneer LX équipé d'un

scanner laser SICK LMS 300 doté d'une portée de 3 m et d'un champ de vue de 120° (Fig. 4.15).



Figure 4.15 – (a) Omron Adept MobileRobots Pioneer LX. (b) SICK LMS 300

Le robot opère dans un environnement intérieur (longueur d'environ 15 mètres), principalement composé de couloirs et de plusieurs salles de bureaux (Fig. 4.16).



Figure 4.16 – (a) Position initiale du Pionner LX dans l'environnement (b) Un autre point de vue de l'environnement

La première position du robot génère un seul nuage de points C_1 avec les gaps associés (Fig. 4.17a). La classification des gaps permet ensuite de sélectionner les "vrais gaps" (Fig. 4.17b). À ce stade, comme le scanner ne compte qu'une seule position, le processus de propagation des gaps n'a pas lieu d'être. Par conséquent, les trois "vrais gaps" du nuage de points C_1 permettent de générer trois candidats HCC (c_1, c_2, c_3) (Fig. 4.17c). Le candidat c_1 sera sélectionné comme prochaine position du robot et sera dorénavant notée NBV_1 (Fig. 4.17d).

Cette position NBV_1 est transmise au module "Robotique" afin de permettre de se déplacer automatiquement dans l'environnement jusqu'à la position souhaitée. Dès

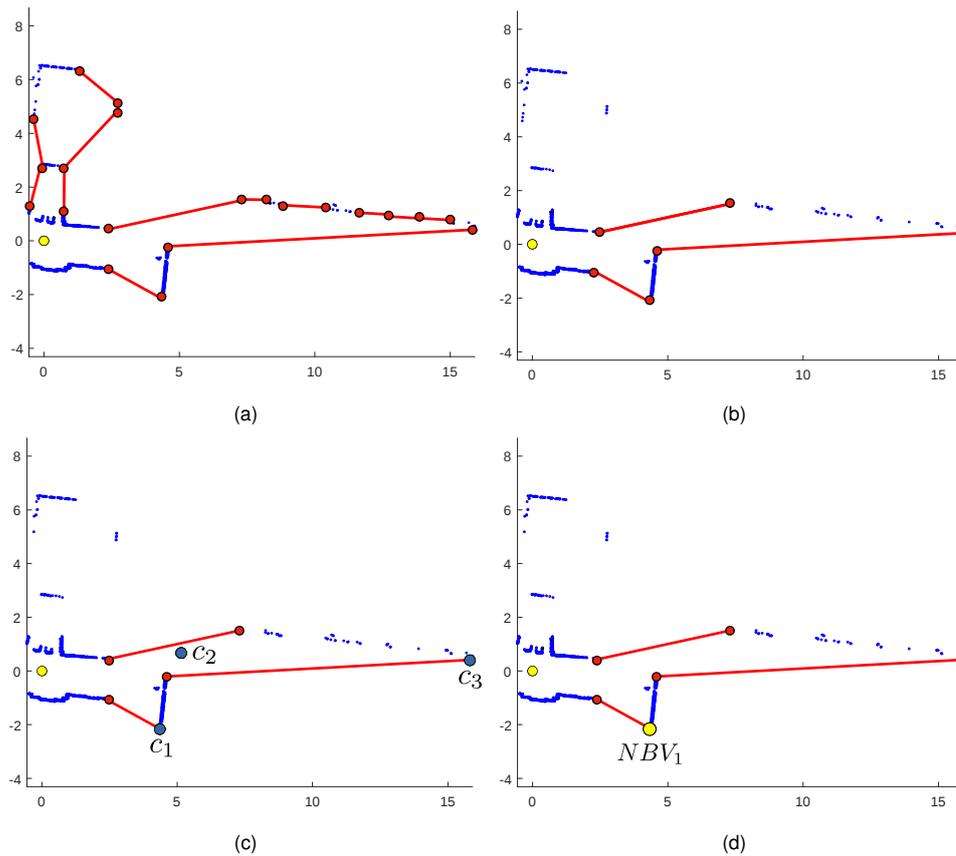


Figure 4.17 – **(a)** Ensemble des gaps détectés dans le nuage de points C_1 avant le processus de classification. **(b)** Ensemble des "vrais gaps" dans le nuage de points C_1 après le processus de classification. **(c)** Candidats générés par notre planificateur de vues (points bleus). **(d)** La NBV sélectionnée.

lors que le robot atteint la position finale, et compte tenu de la limitation de son champ de vision à 120° , il effectue un tour complet sur lui-même pour balayer l'environnement alentour. À la suite de cette opération, un nuage de points C_2 est produit.

Cette deuxième position du scanner nous permet de valider le fonctionnement de la Méthode-Projective. En effet, les nuages de points C_1 et C_2 sont utilisés pour détecter les zones occultées qui restent à explorer (Fig. 4.18), et ainsi à recalculer une nouvelle position NBV_2 .

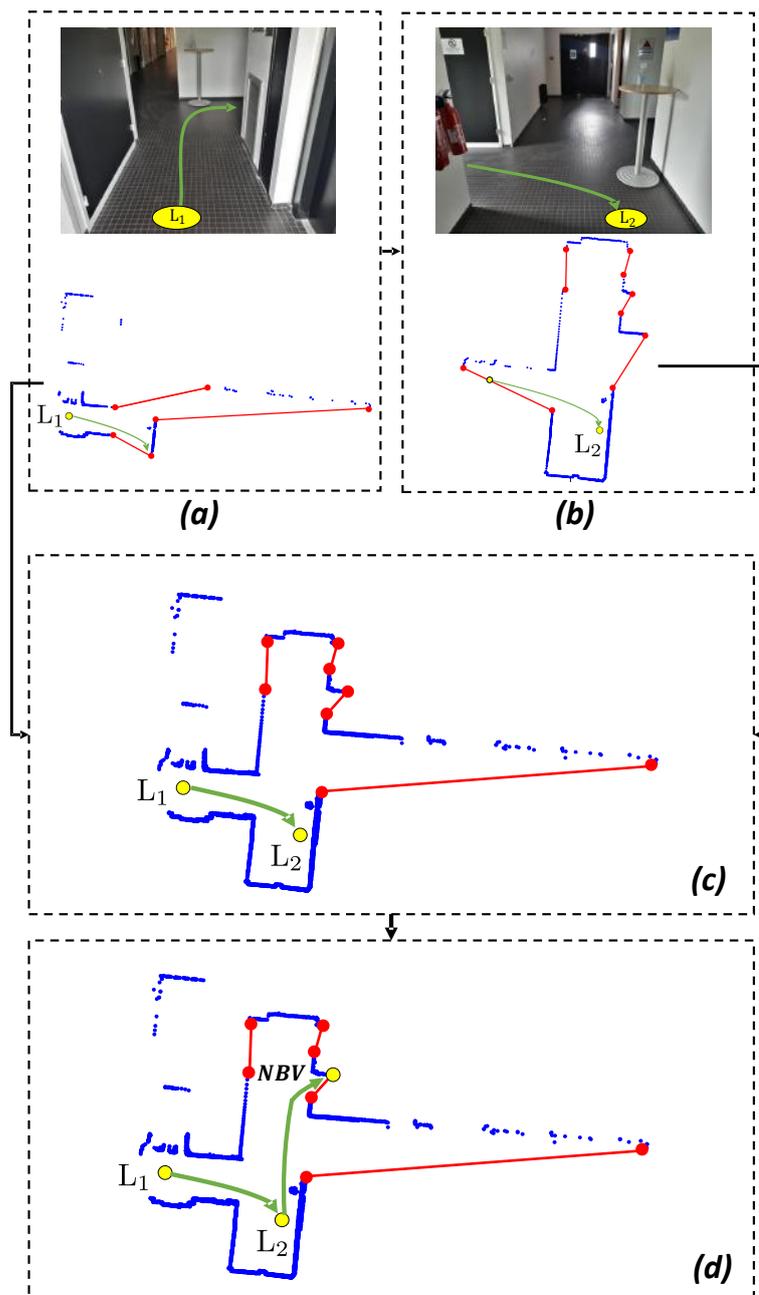


Figure 4.18 – (a) Ensemble des "vrais gaps" dans le premier nuage de points C_1 et (b) dans second nuage de points C_2 après le processus de classification. (c) Détection de l'ensemble des gaps dans le nuage de points global avec la Méthode-Projective. (d) La nouvelle NBV sélectionnée par le planificateur de vues.

Ce processus itératif qui consiste à observer et à décider où se déplacer pour la prochaine position de détection s'arrête lorsqu'il n'y a plus de fragment dont la longueur est supérieure à un seuil prédéfini égale à 0,1 m. La figure 4.19 illustre le chemin final suivi par le robot ainsi que le nuage de points acquis durant le processus d'exploration.

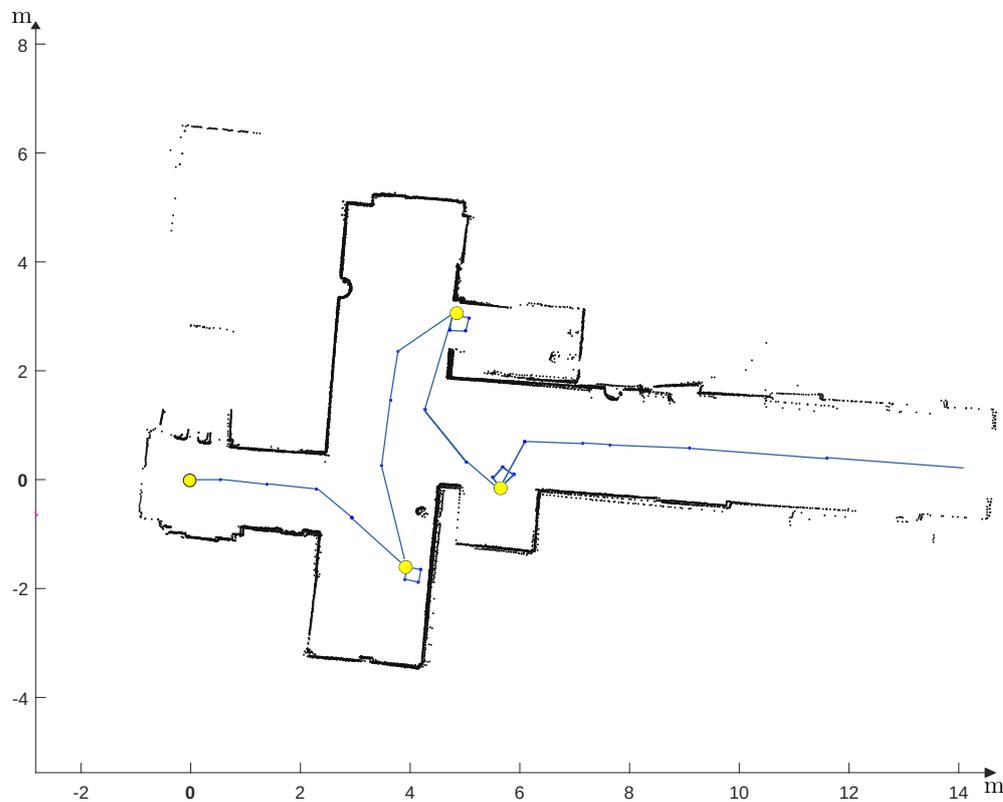


Figure 4.19 – Trajectoire finale calculée en utilisant notre planificateur de vues en Mode 3.

4.5 Conclusion

Dans les chapitres précédents, nous avons proposé un planificateur de vues adapté à une utilisation en Mode 1 avec un scanner laser terrestre, en Mode 2 si l'opérateur humain choisit d'automatiser une partie du processus de numérisation avec le système TLS/Miroir ou en Mode 3 si l'opérateur humain décide d'automatiser complètement la numérisation d'un environnement avec un scanner embarqué sur un robot

mobile.

Dans ce chapitre, nous avons analysé et évalué les performances des trois modes de fonctionnement du planificateur de vues dans des environnements simulés puis réels. La première partie du chapitre détaille le fonctionnement des différentes étapes des trois modes dans un environnement simple avant de présenter des résultats dans des environnements plus complexes. Pour ce faire, nous avons développé un simulateur pour préparer des scénarios, créer des environnements 2D, positionner un scanner laser et générer les nuages de points 2D.

Dans la suite du chapitre, deux études comparatives réalisées avec ce simulateur nous ont permis d'évaluer les performances du planificateur de vues. D'une part, le mode 1 est comparé aux performances d'un opérateur humain expérimenté dans le cas d'un environnement complexe et de grande taille. Il s'agit de l'esplanade de Hassan à Rabat (Maroc), dont la surface est d'environ 2000 m^2 . Les résultats montrent que notre approche permet d'atteindre un taux de couverture similaire à celui de l'opérateur humain avec moins de positions du scanner. D'autre part, le Mode 3 est comparé à quatre autres méthodes de la littérature et démontre que la stratégie proposée est un bon compromis entre la longueur du chemin cumulé par le robot durant son exploration et le temps nécessaire pour scanner l'ensemble de l'environnement.

Finalement, nous avons considéré le cas de l'implémentation de notre approche sur une architecture réelle. Pour ce faire, un robot mobile équipé d'un scanner 2D est utilisé en Mode 3 pour explorer et réaliser un nuage de points complet d'un environnement intérieur. L'expérimentation avec le robot confirme la pertinence de notre approche d'un point de vue pratique.

5.1 Bilan des travaux réalisés

Le présent mémoire a permis de détailler les travaux de recherche qui ont été menés dans le cadre de cette thèse. Le contexte applicatif est celui de la planification optimale des positions d'un scanner laser pour une numérisation complète d'environnements larges et complexes comme la cathédrale notre dame d'Amiens ou la mosquée de Hassan à Rabat. L'objectif principal consistait à réfléchir à la réalisation, la conception et le développement d'un planificateur de vues capable d'assister un opérateur humain avec plusieurs modes de fonctionnement pour s'adapter à une utilisation avec des scanner laser terrestre (TLS) ou des scanners laser embarqués sur des robots terrestres mobiles. Pour ce faire, nous avons d'abord scindé le problème de la planification de vues avec des scanners laser en deux étapes principales : l'analyse de la visibilité et la stratégie d'exploration.

De façon générale, l'analyse de la visibilité consiste à extraire à partir de nuages de points alignés issus de plusieurs positions du scanner une représentation visuelle simple et précise des zones occultées qui restent à numériser. Les zones occultées sont représentées par les frontières qui délimitent le passage entre les zones numérisées et le reste. Cette étape s'avère particulièrement chronophage et complexe avec des méthodes classiques de la littérature lorsqu'il s'agit de nuages de point larges ou complexes. En effet, dans la majorité des travaux, une représentation intermédiaire (grille d'occupation, GNT, Surface de niveaux) est calculée à partir du nuage de points

avant de détecter les frontières. Ainsi, nous avons axé la conception de notre méthode d'analyse de la visibilité sur une représentation simple et précise des frontières directement à partir des nuages de points fournis par le scanner pour réduire au mieux le temps de calcul. Cet objectif est atteint avec la Méthode-Projective qui offre un bon compromis entre la précision et le temps de calcul grâce au pipeline de détection, classification et propagation des gaps.

Après l'analyse de la visibilité vient la phase de la planification des positions optimales du scanner et qui relèvent de la stratégie d'exploration. La littérature dans ce domaine propose plusieurs approches employant différentes stratégies d'exploration pour obtenir la couverture complète de l'environnement à explorer. Chaque stratégie tend à déterminer le chemin le plus court pour explorer l'ensemble de l'environnement en un minimum de temps. Néanmoins, et à l'instar de la phase de l'analyse de la visibilité, la majorité des approches se limite à des petits environnements qui offrent un bon compromis entre le temps de calcul et la longueur du chemin pour explorer l'environnement au complet. Ainsi, l'approche proposée nommée "Cellules Conservatrices Hiérarchiques" permet de réduire considérablement le nombre de candidats pour la sélection de la meilleure position suivante du scanner grâce à des zones spécifiques associées aux frontières (gaps) dans l'environnement que nous avons appelé "Cellules conservatrices". Le nombre réduit des candidats permet de diminuer considérablement le temps de calcul. De plus, l'utilisation d'une fonction d'utilité pour évaluer le score des candidats permet d'éviter un long chemin cumulatif en particulier dans des environnements larges et complexes.

Le planificateur de vues proposé dans le chapitre 2 sous la forme du pipeline "Méthode-Projective + Cellules Conservatrices Hiérarchiques" rend possible l'analyse de la visibilité et le placement optimal du scanner. Ainsi, le choix revient à l'opérateur humain de décider selon les outils à sa disposition d'utiliser le mode "hors-ligne" dans le cas d'un TLS ou en mode "temps réel" dans le cas d'un scanner embarqué sur un robot mobile.

Dans le cas du mode "temps réel", notre planificateur de vues est utilisé avec

un scanner embarqué sur un robot mobile. On considère que le processus est déjà totalement automatisé et qu'il ne nécessite que très peu d'interventions de la part de l'opérateur humain. Par contre, en mode "hors-ligne" avec un TLS, le planificateur de vues se comporte comme un outil d'aide à la décision et permet d'automatiser certaines parties du processus de numérisation et ainsi faciliter la tâche à l'opérateur humain. Néanmoins, le déplacement du TLS reste à la charge de l'opérateur humain, une tâche certes minimisée par le nombre optimal des positions fournies par le planificateur de vues, mais qui reste relativement pénible au vu du poids du TLS. En conséquence, dans le chapitre 3, nous avons proposé d'intégrer une brique supplémentaire au planificateur de vues en mode "hors-ligne" pour automatiser au maximum le processus de numérisation. L'idée qui sous-tend le concept est de déplacer un miroir plan fixé verticalement sur le dessus d'un robot mobile de façon à scanner un maximum de parties occultées de l'environnement grâce à la réflexion des faisceaux laser sur la surface du miroir avant de déplacer manuellement le TLS une nouvelle fois. Le système TLS/Miroir réduit considérablement les déplacements de l'opérateur humain mais nécessite en contrepartie une logistique plus complexe du processus de numérisation.

Nous avons consacré le dernier chapitre de ce mémoire à la partie expérimentale. L'expérimentation sur des données synthétiques fournies et réelles a permis de valider la faisabilité de notre approche en particulier sur des systèmes réels, et sa capacité à résoudre le problème de la planification des vues selon des contraintes préalablement définies. Des tests sur des données synthétiques générées par un simulateur que nous avons développé pour cette tâche spécifique a permis de valider le fonctionnement dans les trois modes de fonctionnement :

- **Mode 1** : Utilisation du planificateur de vues en mode hors-ligne avec un scanner laser terrestre
- **Mode 2** : Utilisation du planificateur de vues en mode hors-ligne avec l'approche basée miroir
- **Mode 3** : Aussi appelé mode "en ligne" ou "temps réel" et il représente l'utilisation du planificateur de vues avec un scanner laser embarqué sur un robot

mobile.

Par la suite, la faisabilité de l'implémentation de notre approche a été validée sur un robot mobile Pioneer LX qui embarque un scanner laser SICK-LIDAR pour explorer et réaliser un scan complet d'un environnement intérieur.

5.2 Quelques perspectives

Plusieurs perspectives et pistes de recherche peuvent être envisagées suite à ce travail. Nous en proposons ici quelques unes pour chacun des modes (Mode 1, 2 et 3) du planificateur de vues.



Figure 5.1 – **(a)** Le système FARO ScanPlan avec un TLS de la même marque **(b)** Application mobile du Leica RTC 360 sur tablette pour le suivi du déplacement du TLS et l'évolution du nuage de points.

Dans le cas du Mode 1, l'extension de la méthode sur une architecture réelle est envisagée. Cette extension ne nous a pas été accessible, car notre appareil TLS (Leica C10) ne fournit pas d'API en temps réel. Toutefois, les dernières versions de TLS comme le Leica RTC 360 ou le système "FARO ScanPlan" (Figure 5.1) offrent désormais des applications mobiles qui permettent à l'utilisateur de suivre le déplacement du scanner et l'évolution du nuage de point en temps réel. Une perspective intéressante est l'intégration de notre approche dans une de ces applications, si les

fabricants de TLS rendent accessibles les données en temps réel.

Dans le cas du Mode 2, l'approche proposée nécessite que le miroir soit dans le champ de vue du scanner pour être scanné. Mais, il est tout à fait envisageable de recourir à plusieurs robots miroir pour numériser davantage de zones occultées sans avoir à déplacer le scanner laser.

Pour le Mode 3, une perspective intéressante est de recourir à plusieurs robots mobiles pour scanner l'environnement. Notre planificateur de vues et notamment la Méthode-Projective peuvent être facilement adaptées à une utilisation multi-robots.

Une autre perspective qui peut faire suite à ce travail est l'extension du planificateur de vues proposé à des nuages de points 3D. Pour ce faire, il sera nécessaire de redéfinir les concepts des gaps, espace visible (scanné) et zones occultées en 3D. Une solution est d'utiliser une représentation en voxels du nuage de points 3D qui sera combinée à une technique de lancer de rayons pour classer les voxels comme obstacle, zone visible ou zone occultée. Les voxels frontières entre les zones visibles et les zones occultées peuvent être assimilés aux gaps en 3D.



BIBLIOGRAPHIE

- Achakir, F., Deseilligny, M., Fkihi, S., Mghari, M., Ettarid, M., Mouaddib, E., & Radgui, A. (2017). The hassan mosque at the digital era. *Frontiers in Science and Engineering International Journal*, 7(1), 45–58.
- Aryan, A., Bosché, F., & Tang, P. (2021). Planning for terrestrial laser scanning in construction : A review. *Automation in Construction*, 125, 103551.
- Ashok, P., Fomin, F. V., Kolay, S., Saurabh, S., & Zehavi, M. (2018). Exact algorithms for terrain guarding. *ACM Transactions on Algorithms (TALG)*, 14(2), 1–20.
- Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor fusion IV : control paradigms and data structures*, vol. 1611, (pp. 586–606). International Society for Optics and Photonics.
- Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., & Siegwart, R. (2016). Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, (pp. 1462–1468).
- Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., & Siegwart, R. (2018). Receding horizon path planning for 3d exploration and surface inspection. *Autonomous Robots*, 42(2), 291–306.
- Biswasa, H. K., Boschéa, F., & Suna, M. (2015). Planning for scanning using building information models : A novel approach with occlusion handling. In *Symposium on Automation and Robotics in Construction and Mining (ISARC 2015)*, vol. 15, (p. 18).

- Boehler, W., Vicent, M. B., Marbs, A., et al. (2003). Investigating laser scanner accuracy. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part 5), 696–701.
- Connolly, C. (1985). The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2, (pp. 432–435).
- Cui, X., & Shi, H. (2012). An overview of pathfinding in navigation mesh. *International Journal of Computer Science and Network Security*, 12(12), 48–51.
- Curless, B., & Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, (pp. 303–312).
- Deng, X., Zhang, Z., Sintov, A., Huang, J., & Bretl, T. (2018). Feature-constrained active visual slam for mobile robot navigation. In *2018 IEEE international conference on robotics and automation (ICRA)*, (pp. 7233–7238).
- Díaz-Vilariño, L., Frías, E., Balado, J., & González-Jorge, H. (2018). Scan planning and route optimization for control of execution of as-designed bim. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- Dornhege, C., & Kleiner, A. (2013). A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6), 459–468.
- Faria, M., Maza, I., & Viguria, A. (2019). Applying frontier cells based exploration and lazy theta* path planning over single grid-based world representation for autonomous inspection of large 3d structures with an uas. *J. Intell. Robot. Syst.*, 93(1-2), 113–133.
- Fasano, A., Callieri, M., Cignoni, P., & Scopigno (2003). Enhancing the 3d laser scanning pipeline using mirrors. *idea*, 510, 900.
- Foulds, L. R. (2012). *Graph theory applications*. Springer Science & Business Media.

- Gao, W., Booker, M., Adiwahono, A., Yuan, M., Wang, J., & Yun, Y. W. (2018). An improved frontier-based approach for autonomous exploration. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, (pp. 292–297).
- González-Banos, H. H., & Latombe, J.-C. (2002). Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, *21*(10-11), 829–848.
- Gutin, G., Yeo, A., & Zverovich, A. (2002). Traveling salesman should not be greedy : domination analysis of greedy-type heuristics for the tsp. *Discrete Applied Mathematics*, *117*(1-3), 81–86.
- Hardouin, G., Moras, J., Morbidi, F., Marzat, J., & Mouaddib, E. M. (2020). Next-best-view planning for surface reconstruction of large-scale 3d environments with multiple uavs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 1567–1574).
- Heidari Mozaffar, M., & Varshosaz, M. (2016). Optimal placement of a terrestrial laser scanner with an emphasis on reducing occlusions. *The Photogrammetric Record*, *31*(156), 374–393.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap : An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, *34*(3), 189–206.
- Jia, F., & Lichti, D. (2018). An efficient, hierarchical viewpoint planning strategy for terrestrial laser scanner networks. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, *4*(2).
- Jia, F., & Lichti, D. D. (2019). A model-based design system for terrestrial laser scanning networks in complex sites. *Remote Sensing*, *11*(15), 1749.
- Jing, W., & Shimada, K. (2018). Model-based view planning for building inspection and surveillance using voxel dilation, medial objects, and random-key genetic algorithm. *Journal of Computational Design and Engineering*, *5*(3), 337–347.

- Käshammer, P., & Nüchter, A. (2015). Mirror identification and correction of 3d point clouds. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), 109.
- Kawashima, K., Yamanishi, S., Kanai, S., et al. (2014). Finding the next-best scanner position for as-built modeling of piping systems. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 45.
- Koch, R., May, S., Koch, P., Kühn, M., & Nüchter, A. (2016). Detection of specular reflections in range measurements for faultless robotic slam. In *Robot 2015 : Second Iberian Robotics Conference*, (pp. 133–145). Springer.
- LaValle, S. M., et al. (1998). Rapidly-exploring random trees : A new tool for path planning.
- Lichti, D. D. (2007). Error modelling, calibration and analysis of an am–cw terrestrial laser scanner system. *ISPRS journal of photogrammetry and remote sensing*, 61(5), 307–324.
- Lichti, D. D. (2008). A method to test differences between additional parameter sets with a case study in terrestrial laser scanner self-calibration stability analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2), 169–180.
- Lin, I.-C., Yeh, J.-S., & Ouhyoung, M. (2002). Extracting 3d facial animation parameters from multiview video clips. *IEEE Computer Graphics and Applications*, 22(6), 72–80.
- Lluvia, I., Lazkano, E., & Ansuategi, A. (2021). Active mapping and robot exploration : A survey. *Sensors*, 21(7), 2445.
- Lopez-Padilla, R., Murrieta-Cid, R., & LaValle, S. M. (2013). Optimal gap navigation for a disc robot. In *Algorithmic Foundations of Robotics X*, (pp. 123–138). Springer.
- Magnusson, M. (2009). *The three-dimensional normal-distributions transform : an efficient representation for registration, surface analysis, and loop detection*. Ph.D. thesis, Örebro universitet.

- Marchand, E., & Chaumette, F. (2017). Visual servoing through mirror reflection. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, (pp. 3798–3804).
- Mariottini, G. L., Scheggi, S., Morbidi, F., & Prattichizzo, D. (2009). Planar catadioptric stereo : single and multi-view geometry for calibration and localization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, (pp. 1510–1515).
- Mobarhani, A., Nazari, S., Tamjidi, A. H., & Taghirad, H. D. (2011). Histogram based frontier exploration. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 1128–1133).
- Monica, R., & Aleotti, J. (2018a). Contour-based next-best view planning from point cloud segmentation of unknown objects. *Autonomous Robots*, 42(2), 443–458.
- Monica, R., & Aleotti, J. (2018b). Surfel-based next best view planning. *IEEE Robotics and Automation Letters*, 3(4), 3324–3331.
- Murphy, L., & Newman, P. (2008). Using incomplete online metric maps for topological exploration with the gap navigation tree. In *2008 IEEE International Conference on Robotics and Automation*, (pp. 2792–2797).
- Myronenko, A., & Song, X. (2010). Point set registration : Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12), 2262–2275.
- Nasir, R., Elnagar, A., et al. (2015). Gap navigation trees for discovering unknown environments. *Intelligent Control and Automation*, 6(04), 229.
- O’rourke, J., et al. (1987). *Art gallery theorems and algorithms*, vol. 57. Oxford University Press Oxford.
- Osher, S., & Fedkiw, R. P. (2001). Level set methods : an overview and some recent results. *Journal of Computational physics*, 169(2), 463–502.

- Papachristos, C., Kamel, M., Popović, M., Khattak, S., Bircher, A., Oleynikova, H., Dang, T., Mascarich, F., Alexis, K., & Siegwart, R. (2019). Autonomous exploration and inspection path planning for aerial robots using the robot operating system. In *Robot Operating System (ROS)*, (pp. 67–111). Springer.
- Papachristos, C., Khattak, S., & Alexis, K. (2017). Uncertainty-aware receding horizon exploration and mapping using aerial robots. In *2017 IEEE international conference on robotics and automation (ICRA)*, (pp. 4568–4575).
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros : an open-source robot operating system. In *ICRA workshop on open source software*, vol. 3, (p. 5). Kobe, Japan.
- Rabbani, T., Dijkman, S., van den Heuvel, F., & Vosselman, G. (2007). An integrated approach for modelling and global registration of point clouds. *ISPRS journal of Photogrammetry and Remote Sensing*, 61(6), 355–370.
- Scott, W. R., Roth, G., & Rivest, J.-F. (2003). View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1), 64–96.
- Selin, M., Tiger, M., Duberg, D., Heintz, F., & Jensfelt, P. (2019). Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robotics and Automation Letters*, 4(2), 1699–1706.
- Senarathne, P., & Wang, D. (2016). Towards autonomous 3d exploration using surface frontiers. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, (pp. 34–41).
- Shen, Z., Song, J., Mittal, K., & Gupta, S. (2017). Autonomous 3-d mapping and safe-path planning for underwater terrain reconstruction using multi-level coverage trees. In *OCEANS 2017-Anchorage*, (pp. 1–6).

- Son, H., Kim, C., & Kim, C. (2015). Fully automated as-built 3d pipeline extraction method from laser-scanned data based on curvature computation. *Journal of Computing in Civil Engineering*, 29(4), B4014003.
- Soudarissanane, S., & Lindenbergh, R. (2011). Optimizing terrestrial laser scanning measurement set-up. In *ISPRS Workshop Laser Scanning 2011, Calgary, Canada, 29-31 August 2011 ; IAPRS, XXXVIII (5/W12), 2011*. International Society for Photogrammetry and Remote Sensing (ISPRS).
- Soudarissanane, S., Lindenbergh, R., Menenti, M., & Teunissen, P. (2009). Incidence angle influence on the quality of terrestrial laser scanning points. In *Proceedings ISPRS Workshop Laserscanning 2009, 1-2 Sept 2009, Paris, France*. ISPRS.
- Soudarissanane, S., Lindenbergh, R., Menenti, M., & Teunissen, P. (2011). Scanning geometry : Influencing factor on the quality of terrestrial laser scanning points. *ISPRS journal of photogrammetry and remote sensing*, 66(4), 389–399.
- Subramani, D. N., & Lermusiaux, P. F. (2016). Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization. *Ocean Modelling*, 100, 57–77.
- Sutantyo, D., Levi, P., Möslinger, C., & Read, M. (2013). Collective-adaptive lévy flight for underwater multi-robot exploration. In *2013 IEEE International Conference on Mechatronics and Automation*, (pp. 456–462).
- Tovar, B., Guilamo, L., & LaValle, S. M. (2004). Gap navigation trees : Minimal representation for visibility-based tasks. In *Algorithmic Foundations of Robotics VI*, (pp. 425–440). Springer.
- Umari, H., & Mukhopadhyay, S. (2017). Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 1396–1402).
- Urrutia, J. (2000). Art gallery and illumination problems. In *Handbook of computational geometry*, (pp. 973–1027). Elsevier.

- Vásquez-Gómez, J. I., López-Damian, E., & Sucar, L. E. (2009). View planning for 3d object reconstruction. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 4015–4020).
- Wang, C., Zhu, D., Li, T., Meng, M. Q.-H., & de Silva, C. W. (2019). Efficient autonomous robotic exploration with semantic road map in indoor environments. *IEEE Robotics and Automation Letters*, 4(3), 2989–2996.
- Wang, Y., Zhang, R., & Qian, L. (2018). An improved a* algorithm based on hesitant fuzzy set theory for multi-criteria arctic route planning. *Symmetry*, 10(12), 765.
- Wujanz, D., & Neitzel, F. (2016). Model based viewpoint planning for terrestrial laser scanning from an economic perspective. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41.
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. Towards New Computational Principles for Robotics and Automation*, (pp. 146–151).
- Yang, S.-W., & Wang, C.-C. (2011). On solving mirror reflection in lidar sensing. *IEEE/ASME Transactions on Mechatronics*, 16(2), 255–265.
- Zhu, C., Ding, R., Lin, M., & Wu, Y. (2015). A 3d frontier-based exploration tool for mavs. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, (pp. 348–352).
- Zuo, X., Yang, F., Liang, Y., Gang, Z., Su, F., Zhu, H., & Li, L. (2020). An improved autonomous exploration framework for indoor mobile robotics using reduced approximated generalized voronoi graphs. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1, 351–359.



LISTE DES PUBLICATIONS

Journal : ISPRS Journal of Photogrammetry and Remote Sensing (Q1, IF (2021) : 8.98)

Titre : Non-Model-Based approach for complete digitization by TLS or mobile scanner

Auteurs : Farouk Achakir, Sanaa El Fkihi, El Mustapha Mouaddib.

Année : 2021

Conférence : ISVC18, 9th International Symposium on Signal, Image, Video and Communications.

Titre : Digitization of the Hassan mosque

Auteurs : Farouk Achakir, Sanaa El Fkihi, Amina Radgui, El Mustapha Mouaddib

Année : 2018

Journal : Frontiers in Science and Engineering International Journal (FSE)

Titre : The Hassan mosque at the digital era.

Auteurs : Farouk Achakir, Marc Deseilligny, Sanaa El Fkihi, Mina Mghari, Mohamed Ettarid, El Mustapha Mouaddib, Amina Radgui

Année : 2017

Résumé: Ce travail de thèse porte sur la numérisation complète d'environnements vastes et complexes en utilisant un TLS ou un scanner mobile. Nous proposons un planificateur de vue multi-objectif adaptatif capable de fonctionner dans un environnement inconnu pour guider l'opérateur humain en mode "hors ligne" et faciliter la tâche de numérisation ou en mode "en ligne" avec un scanner embarqué sur un robot mobile pour une exploration automatique de l'environnement. La méthode proposée suppose que le scanner est déplacé sur une surface plane ce qui est fréquent dans les environnements d'intérieur, les zones urbaines, les espaces ouverts ou encore dans certaines applications du patrimoine culturel.

Dans un premier temps, nous proposons une nouvelle stratégie d'exploration automatisée qui exploite des régions spécifiques de l'environnement, nommées "cellules conservatrices" pour réduire le nombre de déplacements du scanner et obtenir une numérisation complète de l'environnement. Ensuite, nous introduisons une approche pour améliorer le processus de numérisation en mode "hors ligne", en particulier lors de l'utilisation d'un TLS dans les grands environnements. Pour ce faire, nous avons proposé de combiner l'utilisation d'un scanner laser terrestre avec un robot mobile équipé d'un miroir plan. La méthode permet de réduire considérablement l'effort fourni par l'opérateur humain pour déplacer le scanner dans l'environnement et améliore le taux de complétude du nuage de points final. Les méthodes proposées dans cette thèse ont été validées avec des nuages de points simulés et réels avec un TLS et un robot mobile et montrent de bonnes performances en termes de taux de couverture et de temps de calcul par rapport à d'autres stratégies de planification de vues dans la littérature, mais aussi par rapport aux résultats obtenus par un opérateur humain expérimenté dans un environnement large et complexe.

Mots-clés: TLS ; Scanner mobile ; Planification de vues ; Miroir ; Nuage de points; Robotique.

Abstract: This PhD work investigates automatic digitization with complete coverage of large and complex environments using a TLS or a mobile scanner. We propose an adaptive multi-objective view-planner that can operate in an unknown environment to provide in offline mode guidance for the human operator and ease the scanning task or in online mode with a scanner embedded on a mobile robot for an automatic exploration of the environment. The proposed method assumes that the laser scanner is moved on a flat surface which is common in indoor environments, urban areas, open spaces or in various cultural heritage applications.

First, we propose a novel exploration strategy that is completely automated and does not require extensive computations that uses specific regions of the environment called "Conservative-Cells" to drastically reduce the number of sensing positions to achieve complete digitization of the environment. Next, we present an approach to improve the scanning process in "offline" mode, especially when using a TLS in large environments. For this purpose, we suggested combining the use of a terrestrial laser scanner with a mobile robot equipped with a planar mirror. The result is a significant reduction in the effort required by the human operator to move the scanner in the environment and improve the completeness rate of the final point cloud. Proposed methods were validated with simulated and real point clouds on both TLS and mobile robot. The proposed approaches show efficient performance in terms of coverage rate and computational time compared to other view-planning approaches as well as the results of an experienced human operator in a large, complex environment.

Keywords: TLS; Mobile scanner; View planning; Mirror; Point cloud; Robotics