



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE
ET POPULAIRE**

**Ministère de l'Enseignement Supérieur et de
la Recherche Scientifique Université Mohamed
Khider – BISKRA**

**Faculté des Sciences Exactes, des Sciences de la Nature
et de la Vie**

Département d'informatique

N° d'ordre :IVA 02/IVA/M2/2023

Mémoire

Présenté pour obtenir le diplôme de
master académique en

Informatique

Parcours: Image et vie artificielle (IVA)

**Reconstruction d'un environnement dynamique en
utilisant le lidar 2D**

Par :

Allia Bouthaina

Soutenu le 19/06/2023

Devant le jury composé de :

Bentrah Ahlem

MCB

Président

Cherif Foudil

Professeur

Rapporteur

Belounar Saliha

MAA

Examineur

Année universitaire 2022-2023

Résumé

La reconstruction d'un environnement dynamique est une tâche cruciale pour de nombreuses applications liées à la robotique, la sécurité des véhicules autonomes, la surveillance industrielle, la modélisation de l'environnement et la cartographie. Le lidar 2D est largement utilisé pour cette tâche en raison de sa capacité à fournir des mesures précises de distance et d'angle, mais la reconstruction d'un environnement dynamique est complexe en raison des changements rapides et imprévisibles des objets et des agents dans l'environnement. Des algorithmes de traitement de données et des techniques de modélisation avancées sont nécessaires pour assurer une reconstruction précise et en temps réel de l'environnement dynamique. L'objectif principal du travail est de proposer une alternative plus économique à la technologie lidar 3D, utilisant des composants facilement accessibles, mais permettant tout de même la création d'une représentation précise et dynamique de l'environnement.

Mots-clés: Environnement, LIDAR 2D, LIDAR 3D, Robot.

Abstract

The reconstruction of a dynamic environment is a crucial task for many applications related to robotics, autonomous vehicle safety, industrial surveillance, environment modeling, and mapping. LIDAR 2D is widely used for this task due to its ability to provide accurate measurements of distance and angle, but reconstructing a dynamic environment is complex due to the rapid and unpredictable changes of objects and agents in the environment. Advanced data processing algorithms and modeling techniques are required to ensure accurate and real-time reconstruction of the dynamic environment. The main objective of this work is to propose a more economical alternative to LIDAR 3D technology, using easily accessible components, while still allowing for the creation of an accurate and dynamic representation of the environment.

Keywords: Environment, LIDAR 2D, LIDAR 3D, Robot.

ملخص

تعد إعادة بناء بيئة ديناميكية مهمة حاسمة للعديد من التطبيقات المتعلقة بالروبوتات وسلامة المركبات يستخدم ليدار ثنائي الأبعاد على نطاق .المستقلة والمراقبة الصناعية والنمذجة البيئية ورسم الخرائط واسع لهذه المهمة نظرًا لقدرته على توفير قياسات دقيقة للمسافة والزوايا ، ولكن إعادة بناء بيئة ديناميكية خوارزميات معالجة البيانات معقدة بسبب التغيرات السريعة وغير المتوقعة للأشياء والعوامل في البيئة. الهدف .وتقنيات النمذجة المتقدمة مطلوبة لضمان إعادة بناء دقيقة وفي الوقت الحقيقي للبيئة الديناميكية الرئيسي من العمل هو اقتراح بديل أكثر اقتصادا لتقنية ليدار ثلاثية الأبعاد ، باستخدام مكونات يسهل الوصول إليها ، ولكن لا يزال يسمح بإنشاء تمثيل دقيق وديناميكي للبيئة

كلمات البحث: بيئة , ليدار ثنائي الأبعاد , ليدار ثلاثي الأبعاد

Introduction générale

Table de matière

Contents

Résumé	2
Abstract	2
ملخص	3
Table de matière	4
Table des figures	6
Liste des tableaux	7
Introduction générale.....	8
Structure du mémoire	8
Chapitre 1: La robotique.....	10
1. Introduction	10
2. Robotique	10
3. Classification des robots.....	11
3.1. Les robots fixes	12
3.2. Les robots mobiles.....	12
4. Modélisation d'un robot	14
5. Types de robots	15
5.1 Robots industriels.....	16
5.2. Robots chirurgicaux	16
5.3. Robots domestiques.....	18
5.4. Robots militaires.....	18
5.5. Robots explorateurs.....	19
5.6. Robots anthropomorphiques.....	20
6. Applications des robots	22
7. Éléments de robot.....	22
7.1. Les pièces mécaniques	23
7.2. Les pièces électroniques.....	23
7.2.1. Système de contrôle :.....	23
7.2.2. Les actionneurs.....	24
7.2.3. Capteurs.....	25
8. LIDAR.....	26
8.1. LIDAR 2D :	27
8.2. LIDAR 3D :	28
8.3. Comparaisons entre LIDAR 2D et 3D:	29

Introduction générale

9. L'autonomie du robot.....	29
10. Avantages inconvénients et risques.....	30
Conclusion :.....	31
Chapitre 2 : Navigation dans un environnement	32
1. Introduction	32
2. Modélisation de l'environnement	33
3. Modélisation de l'interaction robot-surface.....	34
4. Représentations topologiques de l'environnement.....	35
4.1. Représentations approximatives de l'environnement.....	35
a. Modèles à base de grilles.....	35
b. Cartes de cheminement.....	36
4.2. Représentations exactes de l'environnement.....	37
4.3. Le modèle de champs de potentiels.....	38
5. Planification de chemin	38
5.1. Les algorithmes de parcours de chemin	39
5.1.1. L'algorithme de Dijkstra	39
5.1.2. L'algorithme A*	40
Conclusion.....	42
Chapitre 03 : Reconstruction 3D de l'environnement.....	43
1. Introduction	43
2. Objectif.....	43
3. Conception de notre système.....	43
3.1 Conception générale.....	44
3.2 Conception détaillée.....	44
Conclusion.....	52
Chapitre 04 : Implémentation et résultats.....	53
1. Introduction	53
2. Configuration matérielle.....	53
3. Environnements et outils	55
3.1. L'environnement de développement.....	55
3.1.1. Raspbian	55
3.1.2. Thonny.....	55
3.2. Langage de développement	56
3.2.1. Python.....	56
3.2.2. Les bibliothèques.....	56
4. Implémentation.....	57
a) La première partie : LIDAR 2D	57

Introduction générale

b) Deuxième partie : servo motor.....	60
c) La troisième partie : Support d'appareil	66
Conclusion.....	66
Conclusion générale	67
Limites et lacunes	67
Travail futur.....	67
Références bibliographiques	68

Table des figures

Figure 1.1 : Qu'est-ce qu'un robot	11
Figure 1. 2 : Exemple d'un robot fixe	12
Figure 1. 3:Exemple de robots mobiles.....	13
Figure 1. 4: Classification des robots par environnement et mécanisme d'interaction [4]	14
Figure 1. 5: Composants d'un système robotique [5].....	15
Figure 1. 6: Exemple de robots industriels.....	16
Figure 1. 7 : Exemples de robots chirurgicaux.....	17
Figure 1. 8 : Exemples de robots domestiques	18
Figure 1. 9: Exemples de robots militaires.....	19
Figure 1. 10: Exemples de robots explorateurs	20
Figure 1. 11: Exemples de robots anthropomorphiques	21
Figure 1. 12: Sophia Robot.....	21
Figure 1. 13: Système de contrôle	23
Figure 1. 14: Configuration de base du LIDAR. [17]	27
Figure 1. 16 : LIDAR 2D	28
Figure 1. 17 : LIDAR 3D	28
Figure 2. 1 : Différents types de décomposition de cellules environnementales.....	33
Figure 2. 2 : Exemples de robots mobiles au sol avec différentes configurations cinématiques.....	34
Figure 2. 4 : Modèles à base de grilles	35
Figure 2. 5 : Cartes de cheminement	36
Figure 2. 6 : Triangulation de Delaunay. [20]	37
Figure 2. 7 : Le modèle de champs de potentiels	38
Figure 2. 8 : Un exemple d'algorithme Dijkstra	40
Figure 2. 9: Un exemple d'algorithme A*	41

Introduction générale

Figure 3. 1: Schéma général du système	44
Figure 3. 2: diagramme de flux du système.	45
Figure 3. 3:Le lidar envoie des signaux et les reçoit.	47
Figure 3. 4:système de numérisation Lidar.	48
Figure 3. 5:Lecture des obstacles par LIDAR à l'intérieur de la pièce	49
Figure 3. 6:structure de données d'un voxel	50
Figure 3. 7:Algorithme de voxel	51
Figure 4. 1 : Raspberry Pi 3B+	54
Figure 4. 2 : Servomoteur Futaba S3003.....	54
Figure 4. 3 : Logo Raspbian	55
Figure 4. 4 : L'interface de Thonny	56
Figure 4. 5 : Image montre un environnement réel et l'environnement obtenu avec le lidar 2D	60
Figure 4. 6 : L'image montre l'environnement réel où un obstacle est placé directement devant le lidar, et l'environnement obtenu le compense de différents côtés.	62
Figure 4. 7 : L'image montre l'environnement réel dans lequel trois obstacles sont placés devant, derrière et à côté du LIDAR, et l'environnement obtenu de différents côtés.	63
Figure 4. 8 : Image montrant un environnement réel dans lequel trois obstacles sont placés, nous avons changé leurs emplacements et l'environnement obtenu de différents côtés	64
Figure 4. 9 L'image montre un environnement réel dans lequel nous plaçons deux obstacles directement devant le lidar et l'environnement est obtenu sous différents angles.....	65
Figure 4. 10 : Support d'appareil	66

Liste des tableaux

Tableau 1:comparaison entre LIDAR 2D et LIDAR 3D.....	29
Tableau 2:Mettez les obstacles en groupes	52

Introduction générale

Introduction générale

La reconstruction d'un environnement dynamique est une tâche cruciale pour de nombreuses applications liées à la robotique, la sécurité des véhicules autonomes, la surveillance industrielle, la modélisation de l'environnement et la cartographie. Le lidar 2D est largement utilisé pour la reconstruction de l'environnement en raison de sa capacité à fournir des mesures précises de distance et d'angle, ce qui permet de reconstruire des informations 3D. Le lidar 2D est également moins cher que les solutions lidar 3D, ce qui en fait une option attractive. La reconstruction d'un environnement dynamique est une tâche complexe car les objets et les agents dans l'environnement peuvent changer rapidement et de manière imprévisible. Par exemple, dans une zone de trafic dense, un grand nombre de voitures, de piétons et d'autres obstacles peuvent entrer et sortir de la zone de mesure du lidar à tout moment, ce qui rend la reconstruction de l'environnement difficile. Dans ce travail, nous allons explorer l'utilisation du lidar 2D pour la reconstruction d'un environnement dynamique et discuter des défis et des avancées dans ce domaine en constante évolution. Nous examinerons également les différentes techniques de traitement de données qui sont utilisées pour résoudre les problèmes liés à la reconstruction d'un environnement dynamique. Nous discuterons également des différentes approches de modélisation de l'environnement et des techniques de cartographie utilisées pour améliorer la précision de la reconstruction.

Structure du mémoire

Le mémoire est composé de deux parties :

Partie 1 : Les avancées scientifiques les plus récentes qui fournissent un contexte théorique au mémoire en passant en revue la littérature pertinente dans plusieurs domaines. Elle se compose de deux chapitres.

Chapitre 1: donne un aperçu général des concepts et de la littérature liés au domaine des robots. Il met l'accent sur la classification des robots, les méthodes de modélisation, les types de robots, leurs domaines d'application et leurs composants. Il aborde également la partie la plus importante de ce travail, à savoir le Lidar bidimensionnel.

Chapitre 2 : Il présenterait de l'introduction à la navigation dans l'environnement, de la modélisation de l'environnement, de l'interaction avec la surface du robot, des représentations topologiques de l'environnement et de la planification de trajectoire.

Introduction générale

Partie 2 : Une explication complète de la reconstruction d'un environnement dynamique en utilisant le Lidar bidimensionnel. Elle se compose également de deux chapitres.

Chapitre 3 : Il présente les spécifications du système, d'un point de vue général et détaillé. Ce chapitre se concentre sur la description de la configuration du système et sa compréhension approfondie.

Chapitre 4 : Ce dernier présente les résultats et les conclusions liés à la mise en œuvre du système.

Chapitre 1: La robotique

Chapitre 1: La robotique

1. Introduction

La robotique a été développée bien avant le 18^{ème} siècle, avec l'apparition des premiers automates conçus pour répondre aux besoins de la Révolution industrielle. Lorsqu'un automate est équipé de capteurs qui lui permettent d'interagir avec son environnement, il devient un robot. Cette capacité à être autonome et intelligent est qualifiée d'intelligence artificielle. Le mot "robot" est dérivé du tchèque "robota" qui signifie "travail, tâche ou corvée". L'écrivain tchèque Karel Čapek a introduit le mot en 1920 dans sa pièce de théâtre "Rossum's Universal Robots", qui a été jouée pour la première fois en 1921. Bien que Karel Čapek soit souvent considéré comme l'inventeur du mot, il a lui-même désigné son frère Josef, peintre et écrivain, comme le véritable inventeur. Nous avons souvent l'image d'un robot comme une machine ayant une forme humaine, capable de marcher, de parler et de faire des gestes, mais en réalité, cela est beaucoup plus simple. Un robot est une machine automatique "programmable" qui peut interpréter les informations provenant de son environnement physique afin d'adapter son comportement. Il a la capacité d'interagir avec l'environnement et d'exécuter différentes fonctions en conséquence. [1]

2. Robotique

La robotique est un terme facile à définir, mais sa réelle portée est souvent difficile à appréhender. Elle se réfère à l'ensemble des techniques développées pour concevoir et produire des robots. Cependant, derrière cette définition se cachent des possibilités infinies et un vaste éventail de domaines d'application. Un robot est une machine qui est équipée de capteurs, d'actionneurs et d'un système logique. [2] La robotique englobe donc toutes les techniques nécessaires à la création de ces machines, qu'elles soient réelles ou virtuelles. [2]

Chapitre 1: La robotique

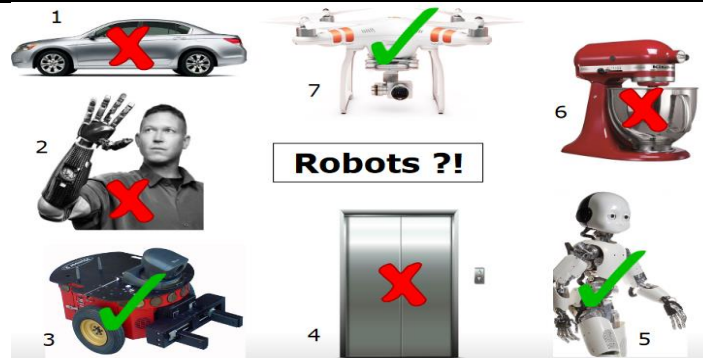


Figure 1.1 : Qu'est-ce qu'un robot

3. Classification des robots

Les robots sont des machines programmables capables d'effectuer des tâches en toute autonomie. Ils peuvent être classés en fonction de l'environnement dans lequel ils évoluent. Cette classification est utile pour comprendre les différentes capacités des robots et les applications pour lesquelles ils sont conçus. La distinction la plus courante dans la classification des robots est basée sur leur mobilité et les distingue en robots fixes et mobiles.

Les robots fixes sont conçus pour travailler dans un environnement immobile et leur position est généralement fixe. Ils sont utilisés dans les usines pour effectuer des tâches répétitives telles que le soudage ou l'assemblage. Les robots fixes sont également utilisés dans les hôpitaux pour assister les chirurgiens pendant les opérations. En revanche, les robots mobiles sont équipés de capteurs et de mécanismes de mouvement pour se déplacer dans leur environnement de travail. Les robots mobiles peuvent être utilisés dans des applications telles que l'exploration spatiale, la surveillance des sites industriels ou la livraison de colis. Les robots mobiles sont divisés en deux catégories principales : les robots terrestres et les robots aériens. Les robots terrestres sont conçus pour se déplacer sur des surfaces planes ou inégales, tels que les robots de livraison de colis. Les robots aériens, quant à eux, sont utilisés pour explorer des zones inaccessibles pour les humains, telles que les zones de catastrophe ou les champs de bataille. La classification des robots en fonction de leur environnement de travail permet de mieux comprendre les capacités et les applications possibles de ces machines.

Les robots fixes et mobiles ont des environnements de travail très différents et nécessitent des capacités différentes pour accomplir leurs tâches spécifiques. La compréhension de cette classification est donc essentielle pour choisir le bon robot pour une application donnée.

Chapitre 1: La robotique

3.1. Les robots fixes

Le terme "robots fixes" désigne les robots qui travaillent dans des environnements définis, généralement dans l'industrie. Ils sont conçus pour effectuer des tâches spécifiques et répétitives, ce qui permet de libérer les travailleurs humains des tâches pénibles et dangereuses. Les robots industriels peuvent être utilisés pour différentes tâches, telles que le soudage, la peinture, l'assemblage et l'emballage dans les usines de fabrication. Cependant, avec l'évolution des technologies de capteurs et des dispositifs d'interaction homme-robot, les manipulateurs robotiques sont de plus en plus utilisés dans des environnements moins contrôlés. Par exemple, ils sont maintenant utilisés dans la chirurgie de haute précision pour effectuer des tâches complexes et délicates sous la supervision d'un chirurgien humain. Les robots fixes sont souvent programmés pour effectuer des tâches précises et ne sont pas mobiles. Ils sont conçus pour opérer dans des espaces limités et ne nécessitent généralement pas de mouvement important. Les robots fixes sont également utilisés dans des environnements dangereux, tels que les centrales nucléaires, où ils peuvent être utilisés pour effectuer des tâches de maintenance sans exposer les travailleurs humains aux radiations nocives. Les robots fixes sont des outils précieux pour l'industrie et la médecine, car ils peuvent effectuer des tâches répétitives et dangereuses avec une grande précision et fiabilité, tout en réduisant les risques pour les travailleurs humains.

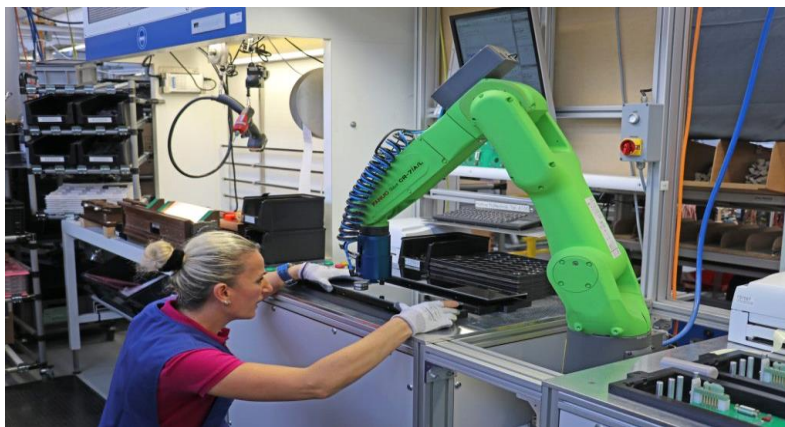


Figure 1. 2 : Exemple d'un robot fixe

3.2. Les robots mobiles

Un robot mobile est une machine complexe équipée de systèmes mécaniques, électroniques et informatiques qui lui permettent d'interagir physiquement avec son environnement. Contrairement aux robots fixes qui sont confinés dans des environnements spécifiques, les robots mobiles sont conçus pour se déplacer et accomplir leur mission dans des environnements variés et souvent imprévisibles. Les robots mobiles peuvent être utilisés dans des domaines tels

Chapitre 1: La robotique

que l'exploration spatiale, la recherche et le sauvetage, la surveillance de la sécurité, la logistique, l'agriculture et bien d'autres encore. Ces machines sont capables d'effectuer une variété de tâches grâce à leurs capacités de perception, de décision et d'action. Les capteurs permettent au robot mobile de recueillir des informations sur son environnement, tandis que les algorithmes d'intelligence artificielle lui permettent de prendre des décisions en temps réel. Enfin, les actionneurs permettent au robot de se déplacer, saisir des objets, ou effectuer d'autres actions. Le robot mobile doit être capable de s'adapter à son environnement en constante évolution, et être capable de faire face à des situations imprévues tout en accomplissant sa mission. Les robots mobiles peuvent être programmés pour agir de manière autonome ou être contrôlés par un opérateur humain à distance. Avec les avancées technologiques rapides. [3]



Figure 1. 3:Exemple de robots mobiles

La distinction entre les robots fixes et les robots mobiles ne sont pas toujours clairement définis, car certains robots peuvent être considérés comme appartenant aux deux catégories en fonction de la manière dont ils sont utilisés. Par exemple, les robots industriels peuvent être équipés de roues ou de rails pour se déplacer dans l'environnement de production. Cependant, il est toujours utile de considérer ces deux classes de robots comme fondamentalement différentes en termes de conception et de fonctionnement. Les robots mobiles, quant à eux, sont conçus pour fonctionner dans des environnements plus variés et peuvent être utilisés pour des tâches telles que l'exploration sous-marine, la surveillance aérienne et les livraisons automatisées. Ils doivent être capables de s'adapter à des situations inattendues et de naviguer dans des environnements complexes, ce qui nécessite des capacités de perception et de décision avancées. De plus, les robots mobiles sont souvent classés en fonction de leur mécanisme de mouvement : aquatique, terrestre ou aérien. Les robots mobiles aquatiques sont conçus pour opérer sous l'eau, que ce soit pour l'exploration scientifique, la surveillance ou la maintenance. Les robots terrestres

Chapitre 1: La robotique

peuvent être équipés de jambes, de roues ou de chenilles pour se déplacer sur une variété de surfaces, et sont souvent utilisés pour la logistique, l'agriculture et la sécurité. Les robots aériens, quant à eux, comprennent des drones et des aéronefs plus lourds que l'air, tels que les avions et les hélicoptères. Ils sont souvent utilisés pour la surveillance, la cartographie et la livraison de colis. En fonction de leur mécanisme de vol, les robots aériens peuvent être classés en aéronefs à voilure fixe ou à voilure tournante. Comme les différents types de robots ont des avantages et des limites en fonction de l'environnement dans lequel ils sont utilisés et des tâches qu'ils sont censés accomplir. La classification des robots en fonction de leur environnement et de leur mécanisme de mouvement peut aider à guider la conception et le développement de robots plus avancés et plus performants. [4]

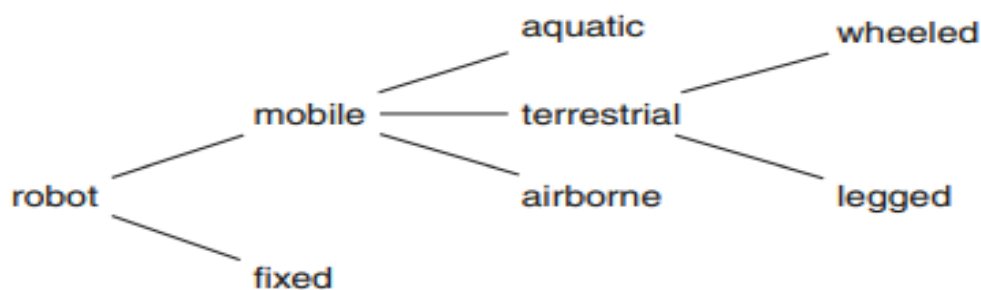


Figure 1. 4: Classification des robots par environnement et mécanisme d'interaction [4]

4. Modélisation d'un robot

La modélisation d'un robot manipulateur est un processus complexe qui peut être réalisé à plusieurs niveaux, en fonction des spécifications de l'application. Cette modélisation permet de concevoir un robot capable d'effectuer des tâches précises et d'interagir avec son environnement de manière sécurisée et efficace. La modélisation d'un robot manipulateur comprend deux types de modèles principaux :

Les modèles géométriques, cinématiques et dynamiques, qui décrivent la structure physique du robot et permettent de générer des mouvements précis en fonction des spécifications de l'application.

Chapitre 1: La robotique

Ces modèles sont essentiels pour la programmation des mouvements du robot et pour assurer une précision optimale.

Les modèles statiques, qui décrivent les interactions entre le robot et son environnement. Ces modèles prennent en compte les forces et les pressions exercées sur le robot et permettent de prédire les comportements du robot en situation réelle. Ces modèles sont donc essentiels pour garantir la sécurité du robot et éviter tout dommage potentiel pour l'environnement.

La modélisation d'un robot manipulateur est une étape clé dans la conception d'un robot efficace et sûr. Les modèles géométriques, cinématiques et dynamiques permettent de programmer des mouvements précis et répétables, tandis que les modèles statiques assurent la sécurité du robot en prévoyant les interactions avec l'environnement. [5]

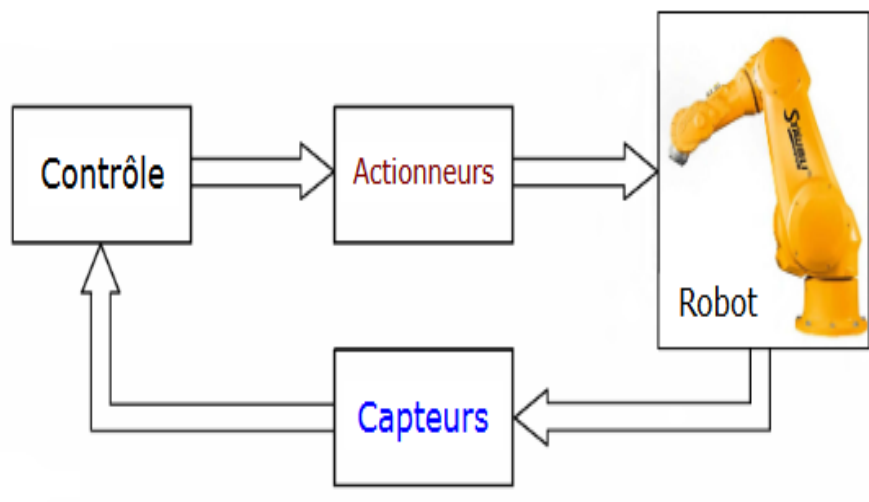


Figure 1. 5: Composants d'un système robotique [5]

5 .Types de robots

Il existe une grande variété de robots conçus pour répondre aux demandes et exigences des consommateurs. Ces robots sont utilisés dans des applications très diverses, allant de la navigation pour donner des directions, au stockage et à la récupération d'objets sur des étagères, en passant par la soudure de pièces métalliques dans des environnements dangereux, et bien plus encore. Malgré la grande diversité des applications robotiques, les robots d'aujourd'hui peuvent être regroupés en six catégories. [6]

Chapitre 1: La robotique

5.1 Robots industriels

Le robot industriel est une machine capable d'effectuer des tâches dès lors que l'on manipule sa programmation. Trois composants le caractérisent : une partie mécanique, une autre électronique, faite de capteurs dans le système de commande, et la dernière informatique qui permet de le relier à l'utilisateur et à son environnement. Ils sont utilisés. Pour de nombreuses missions dans le secteur industriel : assemblage, conditionnement, inspections en zones sensibles, découpe laser, vérifications de pièces. Selon leur niveau de technicité, ils sont plus ou moins multitâches. Du robot cartésien qui ne circule que selon les trois axes X, Y, Z au robot à plusieurs degrés de liberté avec des articulations qui lui permettent d'être fluide dans l'espace mais statique, jusqu'au robot SCARA qui se déplace sur les trois axes tout en ayant un angle de rotation. [7] Les robots industriels sont les premiers à avoir été produits en grand nombre. Ils se trouvent plus particulièrement sur les chaînes de montage, et le plus souvent dans l'industriel automobile. Il existe des robots soudeurs, de démolition, de nettoyage, emballage ou de surveillance. [7]



Figure 1. 6: Exemple de robots industriels

5.2. Robots chirurgicaux

Robots chirurgicaux ils sont en fait une autre branche de robots industriels. Ils fonctionnent soit par bras mécaniques reliés à un ordinateur, soit par bras articulés dirigés par le chirurgien et dont les mouvements sont reproduits en même temps. Il y a environ dix ans, l'apparition d'un robot chirurgical (Da Vinci Intuitive) a révolutionné cette manière de procéder.

Chapitre 1: La robotique

Nous étions toujours en chirurgie minimalement invasive, mais grâce à la technologie robotique, le chirurgien retrouvait une vision tridimensionnelle par l'existence de deux canaux optiques, et il retrouvait une mobilité en sept dimensions, grâce aux instruments articulés, qui s'avérait supérieure à celle de la main et du poignet du chirurgien ! [8] La chirurgie robotique a amené une autre dimension à l'acte chirurgical, puisque pour la première fois le chirurgien se distance de son patient, et il réalise l'intervention assis à une console qui lui permet de diriger à distance la caméra et les instruments. Une réelle interface survient dès ce moment entre le chirurgien et le patient, et il est aisé d'imaginer que cette interface va permettre l'interposition de supports techniques, d'imagerie ou de guidance, qui assisteront et faciliteront le geste chirurgical. [8]

Le développement de la chirurgie robotique, initialement limité à l'urologie, se répand maintenant extrêmement rapidement à la chirurgie viscérale, à la chirurgie thoracique, à la chirurgie cardiovasculaire et surtout à la chirurgie gynécologique. Après avoir démontré la sécurité liée à l'utilisation de ce robot assez complexe techniquement, des travaux toujours plus nombreux démontrent le bénéfice lié à l'utilisation de ce robot. La diminution du nombre et de l'incidence des complications et/ou de leur sévérité, constitue une économie financière qui permet de justifier l'investissement qui doit être consenti à l'achat d'un robot. Dépenser pour la chirurgie robotique, c'est économiser sur les complications opératoires lorsque le robot est utilisé à bon escient.

Il ne s'agit là que d'une explication et d'une argumentation financière à laquelle on peut ajouter l'élément humain d'un patient, dont l'évolution postopératoire est grevée de moins de comorbidités. [8]

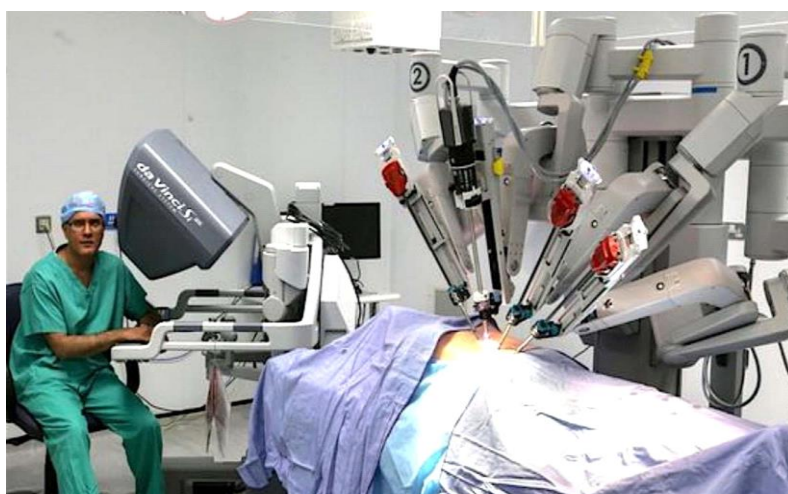


Figure 1. 7 : Exemples de robots chirurgicaux

Chapitre 1: La robotique

5.3. Robots domestiques

Robots domestiques ils peuvent accomplir de multiples tâches ou simplement nous divertir. Plusieurs sociétés ont créés une tondeuse à gazon automatique qui, sur un périmètre donné, peut éviter les obstacles tout en coupant le gazon. Les robots domestiques sont utilisés en vaisselle, en repassage, en nettoyage. Ils peuvent également être utilisés dans le domaine de la restauration Pour nous divertir, une marque très connue, a conçu un robot chien nommé Aibo .La version la plus récente possède des fonctionnalités très avancées telles que appareil photo, reconnaissance vocale. Jusqu'à présent, il n'y a que quelques rares modèles. On estime à 3 540 000 le nombre de robots domestiques en 2006, avec une estimation de 950 000 robots industriels. Les robots domestiques sont utilisés en vaisselle, en repassage, en nettoyage. Ils peuvent également être utilisés dans le domaine de la restauration. [9]



Figure 1. 8 : Exemples de robots domestiques

5.4. Robots militaires

Robots militaires Ils sont principalement utilisés pour la surveillance dans les airs comme dans la mer. Il existe un avion américain sans pilote qui surveille, reconnaît, identifie ou même détruit des cibles ennemies. Il y a également un sous-marin qui a les mêmes propriétés .Le déploiement de robots dans l'armée a révolutionné la manière dont les soldats interagissent sur le terrain de guerre. Aujourd'hui, il existe une variété de robots militaires, chacun avec un rôle spécifique dans la sécurité et la survie des soldats.

Les robots démineurs sont l'un des types les plus courants de robots militaires et sont utilisés pour la détection et la neutralisation des explosifs. Ces robots sont particulièrement utiles dans les zones de guerre où les explosifs sont couramment utilisés. Ils permettent de réduire considérablement le risque pour les humains en limitant le danger d'explosion. Les drones sont

Chapitre 1: La robotique

également largement utilisés dans l'armée pour les missions de reconnaissance et de surveillance. Les drones de reconnaissance sont équipés de caméras et d'autres capteurs pour repérer les ennemis et évaluer les dangers potentiels. Les drones de combat, quant à eux, sont équipés d'armes et peuvent être utilisés pour mener des frappes ciblées. Les robots de combat, tels que le robot SWORDS, sont conçus pour aider les soldats dans les combats. Ils peuvent être équipés d'armes et d'autres équipements pour soutenir les soldats sur le terrain. Le coût élevé de ces robots de combat peut poser un problème, mais leur efficacité en termes de réduction des pertes humaines est incontestable. Les robots peuvent être déployés dans des situations où la sécurité des soldats est mise en danger, ce qui peut permettre de sauver des vies.

En conclusion, les robots militaires sont devenus des outils précieux dans les opérations de guerre modernes. Leur utilisation réduit considérablement le risque pour les soldats, leur permettant ainsi de remplir leur mission avec un minimum de danger. Cependant, leur développement et leur déploiement nécessitent une réflexion éthique approfondie pour garantir que leur utilisation est juste et respectueuse des droits de l'homme. [10]



Figure 1. 9: Exemples de robots militaires

5.5. Robots explorateurs

Les robots explorateurs est un robot qui permet de par sa taille ou ses capacités particulières d'explorer de nouveaux environnement. Cela permet parfois de limiter les pertes humaines. Les robots explorateurs remplacent l'homme dans des environnements difficiles. L'exploration de la centrale nucléaire de Tchernobyl a été faite par un robot. L'exploration de l'espace se fait de

Chapitre 1: La robotique

plus en plus par des robots. En 1976, la sonde Viking 1 s'est posée sur Mars et a pris les premières photos de la planète. [11]

Avantage de l'utilisation des robots explorateurs ?

L'utilisation de robots explorateurs est parfois la seule solution pour accéder à certains lieux. En effet les robots explorateurs ont de nombreux avantages :

- Leurs tailles par exemple qui n'étant pas humaine leur permettent souvent de se faufiler dans des environnements non accessibles à l'homme.
- Leurs capacités à s'adapter à des milieux extrêmes, comme dans l'eau ou dans des milieux aux températures très froides ou au contraire très chaudes.
- Ces robots sont utilisés pour résoudre des affaires comme par exemple les naufrages de bateaux mais encore pour explorer des territoires ennemis. En envoyant ces robots, on limite les captures d'hommes et donc les risques de pertes humaines.
- Ces robots peuvent simplement être utilisés pour explorer des reliefs terrestres inconnus ou même des autres planètes. [11]



Figure 1. 10: Exemples de robots explorateurs

5.6. Robots anthropomorphiques

Les robots anthropomorphiques qui ressemblent à l'être humain, sont une des principales motivations des roboticiens. Il existe deux types de robots humanoïdes :

Le premier type est constitué de robots conçus pour effectuer des tâches que seuls les humains étaient capables d'accomplir jusqu'à présent. De plus, ils sont programmés pour obéir aux ordres et interagir avec les humains. L'un des plus grands développements technologiques dans le domaine des robots est probablement le robot " ASIMO", développé par Honda au Japon et

Chapitre 1: La robotique

introduit en 2000. Il mesure 4 pieds et 3 pouces de hauteur et est conçu pour fonctionner dans des environnements humains, en utilisant ses jambes pour marcher et courir, et ses bras pour effectuer des tâches telles que le transport d'objets. ASIMO est équipé d'un ensemble de capteurs et de caméras, qui lui permettent de reconnaître et de répondre aux personnes et à son environnement, et d'effectuer des tâches de réceptionniste ou de guide d'information. Ce robot de recherche devrait finalement être capable d'aider les personnes handicapées, les personnes âgées ou les patients. [12]



Figure 1. 11: Exemples de robots anthropomorphiques

Le deuxième type, robot " Sophia" développé par "Hanson Robotics" à Hong Kong et introduit pour la première fois en 2016. Il est conçu à l'image d'Audrey Hepburn et conçu pour ressembler et agir comme un être humain. Sophia a été programmée pour utiliser des expressions faciales et des gestes pour communiquer avec les gens et est capable de tenir une conversation sur une variété de sujets. Il utilise l'intelligence artificielle et les algorithmes d'apprentissage automatique pour améliorer constamment ses réponses. [13]



Figure 1. 12: Sophia Robot

Bien que "ASIMO" et "Sophia" soient tous deux des robots humanoïdes, ils ont des objectifs et des capacités de conception différents. ASIMO est conçu pour fonctionner dans des

Chapitre 1: La robotique

environnements humains et effectuer des tâches physiques, tandis que Sophia est conçue pour interagir avec les gens et tenir des conversations.

6. Applications des robots

L'utilisation des robots a connu une évolution considérable au fil du temps. Les robots ont été initialement conçus pour effectuer des tâches répétitives dans les industries, mais leur utilisation s'est étendue à d'autres domaines. De nos jours, les robots sont omniprésents dans notre vie quotidienne, que ce soit dans un contexte domestique ou professionnel. Dans le domaine de la médecine, les robots sont utilisés pour réaliser des examens, des traitements (tels que la radiothérapie) et des interventions chirurgicales dans des zones difficiles d'accès qui requièrent une grande précision. Cette technologie offre des avantages considérables en termes de précision et de sécurité pour les patients. Dans le domaine militaire, les drones sont utilisés pour survoler des zones de conflit, transporter du matériel ou lancer des missiles. Les robots sont également utilisés dans le domaine spatial pour explorer l'espace, la lune ou des planètes. Dans un contexte domestique, les robots sont utilisés pour effectuer des tâches ménagères, comme les tondeuses et les aspirateurs, libérant ainsi du temps pour les occupants de la maison. Les robots sont également utilisés dans le domaine artistique, comme les robographes pour réaliser des dessins.

Les robots sont également utilisés dans l'industrie automobile et agroalimentaire pour accroître l'efficacité et la précision des processus de production. La sécurité est un autre domaine où les robots sont utilisés, notamment pour la vidéosurveillance. Les robots sont utilisés dans le domaine du transport pour transporter des personnes ou des marchandises. En somme, les robots sont omniprésents et ont trouvé de nombreuses applications dans différents secteurs, offrant des avantages considérables en termes de précision, d'efficacité et de sécurité.

7. Éléments de robot

Un robot est composé d'une combinaison de pièces mécaniques et électroniques, qui sont contrôlées par un microprocesseur. S'il est mobile, le robot doit également disposer d'une source d'énergie embarquée, qui est généralement une batterie rechargeable. Les robots peuvent être conçus pour fonctionner de manière autonome, c'est-à-dire sans intervention humaine directe, ou être contrôlés à distance par un opérateur humain. Les robots autonomes sont souvent équipés de capteurs qui leur permettent de percevoir leur environnement et de prendre des

Chapitre 1: La robotique

décisions en conséquence. Les capteurs peuvent inclure des caméras, des microphones, des capteurs de pression, des capteurs de mouvement et bien d'autres encore. [14]

7.1. Les pièces mécaniques

Les pièces mécaniques permettent de bâtir une ossature du robot qui sera pilotée par le microprocesseur et les composants électronique qui agiront comme des capteurs ou des actionneurs pour faire agir voire réagir le robot. C'est la partie la plus basique du robot, qui comprend le châssis, la roue et le placement. [14]

7.2. Les pièces électroniques

Les microprocesseurs sont des éléments primordiaux d'un robot, car ils permettent l'exécution de logiciels informatiques donnant son autonomie au robot. On trouve souvent dans un robot des modèles à très faible consommation, notamment pour des robots de petite taille, qui ne peuvent pas emporter avec eux une source d'énergie importante.

Les moteurs permettent à des éléments mobiles de bouger suivant un ou plusieurs degrés de liberté. Les capteurs ultrason permettent de détecter les obstacles et le plus souvent de les éviter!

Les caméras sont les yeux des robots. Il en faut au moins deux pour permettre la vision en trois dimensions. Le traitement automatique des images pour y détecter les formes, les objets, voir les visages, demande un général un traitement matériel car les microprocesseurs embarqués ne sont pas assez puissants pour le réaliser.

7.2.1. Système de contrôle :

Le cerveau ou la carte contrôleur est comme un petit ordinateur. Il est en fait un microcontrôleur qui peut être connecté à des capteurs et à des actionneurs (muscles, voix, etc...). [1]

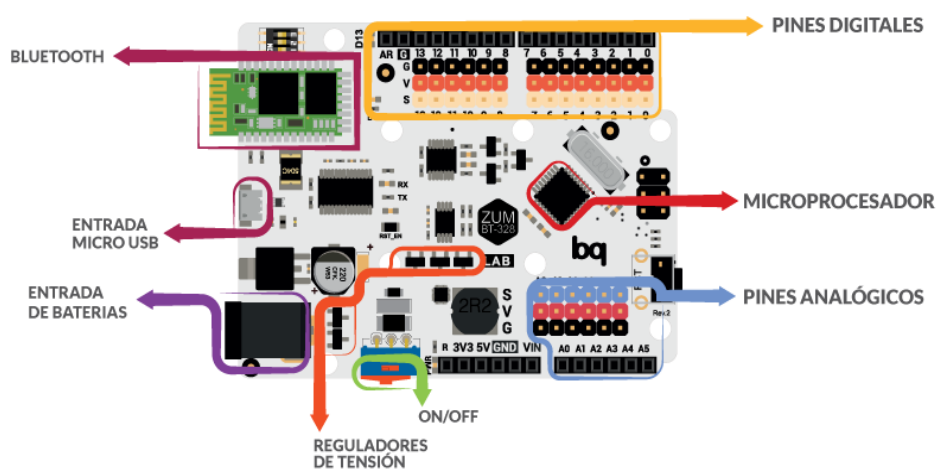


Figure 1. 13: Système de contrôle

Chapitre 1: La robotique

La carte ZUM est composée des éléments suivants :

- Microprocesseur: La micropuce effectue toutes les opérations nécessaires pour rendre votre carte opérationnelle avec votre programme.
- Entrées numériques: Elles ne peuvent avoir que deux valeurs: 0 ou 1 (tout ou rien, On ou Off).
- Entrées analogiques: Elles peuvent avoir des valeurs multiples: 0, 1, 34, 255, etc....
- Entrée Micro-USB: La carte ZUM communique avec votre ordinateur via ce port.
- Entrée de la batterie: La carte électronique reçoit l'énergie des batteries ou de l'unité d'alimentation de sorte qu'elle puisse fonctionner.
- Bluetooth: Il permet à la carte électronique de communiquer sans fil avec un ordinateur, un smartphone ou d'autres appareils.
- Bouton ON/OFF: Il coupe l'alimentation de votre carte électronique de sorte que vous pouvez programmer le robot sans risque que celui-ci ne se mette à fonctionner.
- Régulateurs de tension: Ils adaptent la tension d'alimentation pour la rendre égale à celle de la carte électronique : 5V. [1]

7.2.2. Les actionneurs

Un actionneur est un composant électronique qui peut agir sur le milieu environnant. Par exemple, un moteur qui déplace une roue, une diode électroluminescente qui indique que l'appareil est sous tension, un haut-parleur qui émet un son d'avertissement, etc...

En utilisant une carte électronique ZUM ou similaire, les actionneurs que nous utilisons seront connectés à une broche numérique. Certains actionneurs, tels que des LED, ne peuvent avoir que deux états, tandis que d'autres, tels que des servomoteurs, peuvent avoir plusieurs fonctions, ce qui en fait de composants analogiques. Pour ce type d'actionneur, notre carte électronique, qui se trouve être numérique, fera usage d'une technique couramment utilisée pour synthétiser des signaux continus, appelée modulation de largeur d'impulsions ou MLI. Ne vous préoccupez pas de ce sujet pour l'instant, nous allons y revenir plus tard. [14]

Chapitre 1: La robotique

7.2.3. Capteurs

Un robot se compose généralement de plusieurs capteurs. Ces capteurs aident les robots à prendre des informations et à effectuer une action en conséquence. Les capteurs sont classés comme proprioceptifs ou extéroceptifs, et les capteurs extéroceptifs sont en outre classés comme actifs ou passifs. Un capteur proprioceptif mesure quelque chose d'interne au robot lui-même. L'exemple le plus connu est le compteur de vitesse d'une voiture qui mesure la vitesse de la voiture en comptant les rotations des roues. [1]

Un capteur extéroceptif mesure quelque chose d'extérieur au robot, comme la distance à un objet. [1] Un capteur actif affecte l'environnement généralement en émettant de l'énergie : un télémètre sonar sur un sous-marin émet des ondes sonores et utilise le son réfléchi pour déterminer la distance. Un capteur passif n'affecte pas l'environnement : une caméra enregistre simplement la lumière réfléchie par un objet. Les robots utilisent invariablement certains capteurs extéroceptifs pour corriger les erreurs pouvant provenir des capteurs proprioceptifs ou pour tenir compte des changements de l'environnement. [1]

Un capteur est simplement un composant électronique qui envoie une quantité plus ou moins grande d'électricité selon un élément physique, comme la lumière, le son ou la distance, etc... Pour connecter un capteur à votre carte électronique, vous devez d'abord vérifier si le capteur est numérique ou analogique. [1]

Un capteur numérique

Un capteur numérique ne peut avoir que deux valeurs: 0 ou 1, tout ou rien. Par exemple, un bouton est un capteur numérique, qui peut soit avoir la valeur 1, soit avoir la valeur 0 lorsque l'on appuie ou non dessus. Sur la carte électronique ZUM ou similaire, les capteurs numériques seront connectés sur les broches numériques D0 à D13. [1]

Un capteur analogique

Un capteur analogique peut avoir de multiples états et il est capable de transformer la quantité de lumière, la température ou autres éléments physiques en une valeur comprise entre 0 et 1023. Pour donner un exemple avec un capteur de lumière, c'est qu'il donne une valeur plus petite ou plus grande par rapport à la quantité de lumière reçue. Sur une carte électronique ZUM ou similaire, les capteurs analogiques seront connectés sur les broches analogiques A0-A5. [1]

Chapitre 1: La robotique

8. LIDAR

LIDAR (Light Detection and Ranging) est une technologie qui utilise des capteurs pour émettre des faisceaux lumineux et mesurer le temps qu'il faut à la lumière réfléchiée pour revenir au récepteur. Cela permet de collecter des informations précises sur la distance entre les objets cibles et la source lumineuse, ainsi que sur leur vitesse.

Les systèmes LIDAR peuvent être classés en fonction du nombre de faisceaux lumineux et du mode de balayage de la surface. On trouve des systèmes LIDAR en 1D, 2D ou 3D.

Cette technologie permet de générer des images en 3D ou 2D avec des données spatiales et de profondeur, ce qui permet de détecter, mesurer, localiser et suivre des objets. Elle a révolutionné l'automatisation dans de nombreuses industries, allant de l'automobile à la logistique et à l'aérospatiale.

Dans les environnements industriels tels que les usines et les entrepôts, des capteurs LIDAR en temps réel à 360° sont utilisés pour détecter les obstacles. Dans les véhicules, ces systèmes permettent de détecter les distances et d'éviter les collisions avec les personnes, les objets ou les autres véhicules.

Les capteurs LIDAR installés sur des bras de robot ou des chariots élévateurs à fourche offrent une efficacité, une précision et une sécurité accrue en simplifiant et en optimisant les opérations de manutention, de transport, de chargement et de déchargement, ainsi que le stockage des matériaux.

La société SICK a été pionnière dans la technologie LIDAR dans les années 1980 et a continué à innover dans ce domaine au cours des dernières décennies, contribuant ainsi à façonner le monde de demain dès aujourd'hui.

En résumé, le LIDAR est une technologie puissante qui utilise des faisceaux lumineux pour collecter des informations précises sur la distance et la vitesse des objets. Son application s'étend à de nombreuses industries et continue d'évoluer pour façonner le monde de demain.

[15] [16]

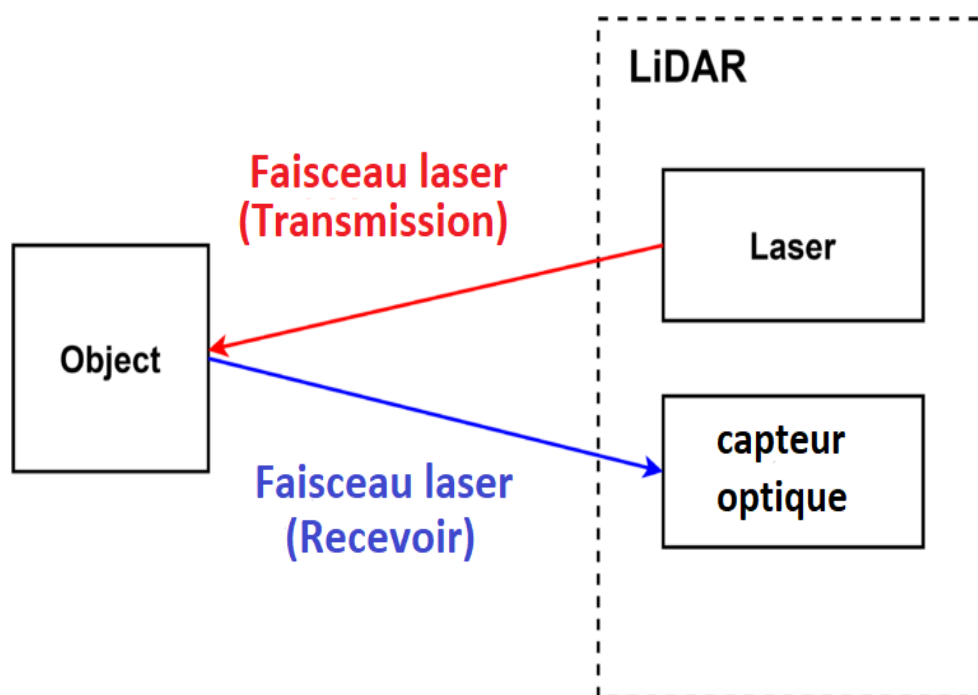


Figure 1. 14: Configuration de base du LIDAR. [17]

8.1. LIDAR 2D :

Le sous-ensemble de capteurs LIDAR qui fonctionnent en deux dimensions utilisent tous un seul plan de lasers pour capturer les dimensions X et Y. Ces données de mesure permettent une navigation et une détection précises des objets, à l'intérieur comme à l'extérieur, quelle que soit la lumière ambiante. Pour une large gamme d'applications, en particulier dans le cadre industriel, cette forme de scanner laser fournit une solution de télémétrie économique. [16]

LIDAR 2D est conçu pour émettre un seul faisceau de lumière vers l'objet cible sur un plan horizontal afin de collecter des données objectives sur les axes X et Y. Habituellement, le capteur LIDAR tourne pour collecter suffisamment d'informations sur les coordonnées X et Y. Entre autres caractéristiques, la taille d'un capteur LIDAR 2D facilite la portabilité par rapport à leurs homologues 3D. Les capteurs LIDAR 2D sont les mieux adaptés pour effectuer des activités de détection. [15]



Figure 1. 15 : LIDAR 2D

8.2. LIDAR 3D :

Les capteurs LIDAR 3D ajoutent une autre dimension au plan de balayage, fournissant ainsi des données de mesure dans un total de trois dimensions. Les données de mesure collectées cartographient numériquement le monde réel, permettant la détection d'objets dans un espace tridimensionnel, même dans des conditions météorologiques extrêmes et dans des environnements indéfinis. Les capteurs LIDAR 3D sont la solution idéale pour de nombreuses applications, même dans des conditions difficiles, du comptage de personnes dans les aéroports au guidage automatique des véhicules, en passant par la localisation précise et l'évitement des collisions, par exemple dans les ports, les centres logistiques et les usines. [16]

LIDAR 3D il utilise des types particuliers de capteurs qui tournent à 360 degrés émettant plusieurs faisceaux de lumière vers le plan vertical d'un objet pour recueillir les coordonnées X, Y et Z de la cible. La technologie 3D convient à la cartographie et à la numérisation de paysages. [15]



Figure 1. 16 : LIDAR 3D

Chapitre 1: La robotique

8.3. Comparaisons entre LIDAR 2D et 3D:

Les capteurs LIDAR peuvent être classés en deux principales catégories : LIDAR 2D et LIDAR 3D. Voici quelques comparaisons entre ces deux types de LIDAR : [15]

	Nombre de faisceaux lumineux émis	L'axe de focalisation lors de l'émission du faisceau lumineux	Coût	Applications industrielles	Facilité de portabilité
LIDAR 2D	n'émet qu'un seul faisceau de lumière vers une surface	uniquement sur ses axes X et Y, le faisceau de lumière n'est projeté que le long du plan horizontal de l'objet.	pas cher	bien adapté lors de la conduite d'activités qui nécessitent beaucoup de télémétrie et de détection en raison des données horizontales détaillées qu'il peut capturer.	petite taille, Son emplacement peut être facilement modifié
LIDAR 3D	d'émettre plusieurs faisceaux de lumière pour collecter des données d'objet plus détaillées.	concentre son faisceau le long du plan vertical de sa cible pour capturer des données tridimensionnelles sur les axes X, Y et Z.	chère	conviennent le mieux à la cartographie et à la numérisation de paysages en raison du volume de données que ce modèle peut capturer sur l'axe Y d'un objet.	grande taille

Tableau 1 : Comparaison entre LIDAR 2D et LIDAR 3D

9. L'autonomie du robot

Réaliser des robots capables de réagir seuls à l'environnement, c'est-à-dire à un certain imprévu. C'est ce plus ou moins grand degré d'autonomie (permis par une intelligence artificielle) qui rapproche les robots des systèmes complètement autonomes envisagés par la science-fiction et la recherche de pointe. [14]

Une certaine capacité d'adaptation à un environnement inconnu peut, dans les systèmes semi-autonomes actuels, être assurée pourvu que l'inconnu reste relativement prévisible : l'exemple déjà opérationnel du robot aspirateur en est une bonne illustration : le logiciel qui pilote cet appareil est en mesure de réagir aux obstacles qui peuvent se rencontrer dans une habitation, de les contourner, de les mémoriser. Il sauvegarde le plan de l'appartement et peut le modifier en cas de besoin. Il retourne en fin de programme se connecter à son chargeur. Il doit donc fournir une réponse correcte au plus grand nombre possible de stimulations, qui sont autant de données entrées, non par un opérateur, mais par l'environnement. [14]

Chapitre 1: La robotique

L'autonomie suppose que le programme d'instructions prévoit la survenue de certains événements, puis la ou les réactions appropriées à ceux-ci. Lorsque l'aspirateur évite un buffet parce qu'il sait que le buffet est là, il exécute un programme intégrant ce buffet, par exemple les coordonnées X-Y de son emplacement. Si ce buffet est déplacé ou supprimé, le robot est capable de modifier son plan en conséquence et de traiter une zone du sol qu'il ne prenait pas en compte jusqu'alors. [14]

10. Avantages inconvénients et risques

Les avantages :

- L'utilisation de robot réduit les opérations à cœur ouvert grâce aux techniques mini-invasives.
- Elle permet aussi de fonctionner par des incisions plus petites, car la miniaturisation des éléments chirurgicaux des robots et l'articulation des bras engendrent une réduction de la taille des incisions.
- Elle permet aussi de réduire les pertes sanguines, une diminution de douleur, une diminution de risque d'infection.
- De plus, l'utilisation de robot élimine les tremblements de la main grâce à un filtre électronique. [18]

Les inconvénients et les risques :

- D'abord savoir que l'utilisation de robot en chirurgie a très peu de risque.
- Les mouvements ne sont pas complètement autonomes, car ils sont sous l'ordre de l'action du chirurgien.
- Il peut y avoir des risques tels qu'une panne ou une difficulté technique.
- Les inconvénients sont le fait que la technologie robotique est extrêmement chère en termes de frais financiers. [18]

Chapitre 1: La robotique

Conclusion :

Bien que le lidar 3D soit le mieux adapté à la cartographie et à l'analyse détaillée pour une utilisation dans les conceptions techniques, son prix élevé le rend difficile à obtenir et à utiliser pour la plupart des applications. Donc, nous suggérons l'utilisation d'un lidar 2D pour la création d'un environnement dynamique. Cette solution alternative permet de fournir des informations en temps réel sur l'environnement de manière plus abordable, tout en offrant une précision suffisante pour la navigation et la détection d'obstacles. Ça peut être utilisé pour diverses applications, notamment la navigation de robots mobiles, la sécurité des véhicules autonomes et la surveillance de l'environnement industriel.

Chapitre 2 : Navigation dans un environnement

Chapitre 2 : Navigation dans un environnement

1. Introduction

Les robots mobiles sont de plus en plus utilisés dans diverses industries telles que la logistique, l'industrie manufacturière, la santé et l'agriculture. Pour qu'ils puissent être omniprésents, il est essentiel qu'ils soient capables de naviguer de manière sûre et efficace dans des environnements dynamiques. Cependant, c'est un défi complexe car les robots doivent être en mesure de traiter une grande quantité d'informations en temps réel et de prendre des décisions rapides pour éviter les obstacles et atteindre leur destination. La complexité de la navigation est due à la dimension temporelle ajoutée dans l'espace de recherche.

En effet, les robots doivent prendre en compte les mouvements des autres agents dynamiques dans leur environnement, tels que les humains, les autres robots ou les véhicules. Les interactions entre ces agents sont subtiles et imprévisibles, ce qui rend leur modélisation extrêmement difficile. Pour répondre à ces défis, les robots mobiles sont équipés de nombreux équipements intelligents tels que des capteurs, des caméras, des lidars et des GPS, qui leur permettent de modéliser l'environnement, de localiser leur position, de contrôler leur mouvement, de détecter les obstacles et de les éviter en utilisant des techniques de navigation avancées.

La planification sûre du chemin est une fonction clé de la navigation. Les robots doivent être en mesure de détecter les obstacles et de les éviter en temps réel tout en atteignant leur destination. Par conséquent, la sélection judicieuse de la technique de navigation est essentielle dans la planification de la trajectoire d'un robot, en particulier lorsqu'il travaille dans des environnements simples ou complexes.

En résumé, la navigation des robots mobiles est un domaine en constante évolution avec des avancées technologiques continues. Cela ouvre la voie à de nouvelles applications et de nouvelles possibilités pour améliorer l'efficacité et la sécurité des robots mobiles dans différents domaines.

Chapitre 2 : Navigation dans un environnement

2. Modélisation de l'environnement

Les robots mobiles de surface se déplacent d'une position à une autre dans une certaine région de l'espace. Par conséquent, il est nécessaire de considérer comment le modèle de locomotion interagira avec cette surface et comment le planificateur de trajectoire en prendra soin. Par exemple, certains algorithmes nécessitent la construction d'un graphe qui représente en quelque sorte l'environnement dans lequel le robot se déplace. C'est principalement le cas pour les algorithmes de recherche de graphes, qui font partie de la catégorie C-Space Search. Les algorithmes évolutifs tels qu'Ant **Colony Optimizers** (ACO) peuvent également utiliser un graphe. Cet atout peut représenter la façon dont les caractéristiques du terrain qui affectent la navigation du robot sont disposées spatialement dans le scénario. Le graphe en question est supposé ici être construit sur des cartes métriques, reconnaissant l'existence d'autres types de cartes telles que les cartes topologiques et sémantiques. Il existe plusieurs façons de construire un graphe, comme le montre la figure 2-1. [19]

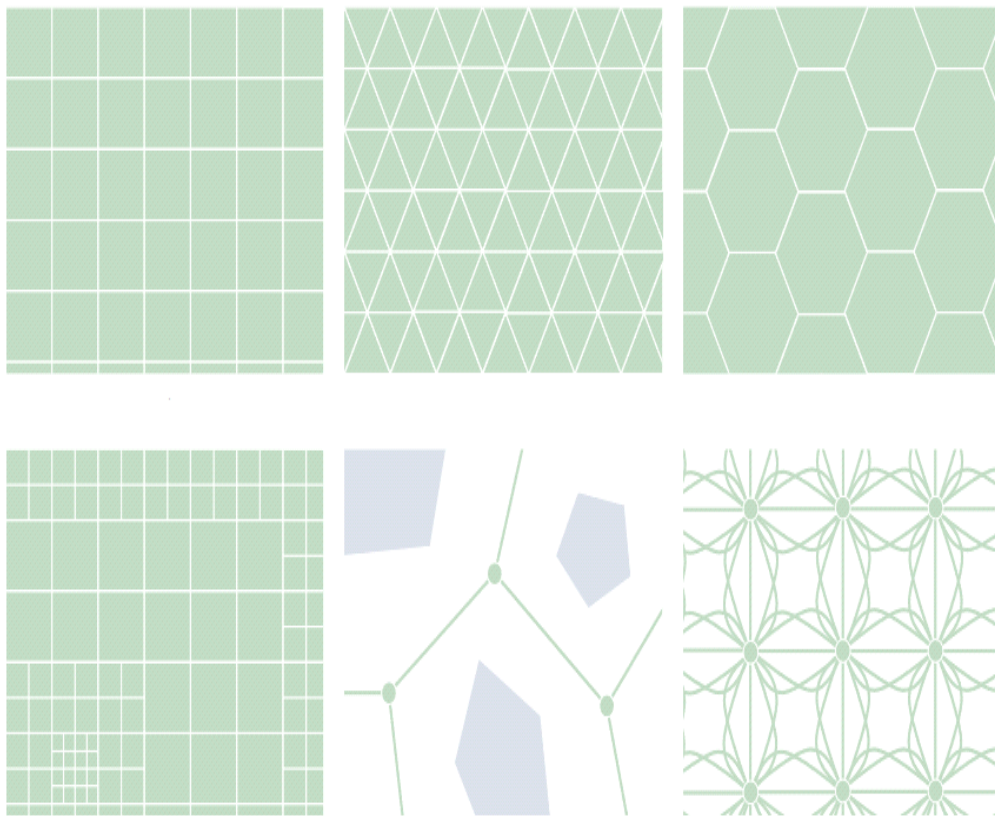


Figure 2. 1 : Différents types de décomposition de cellules environnementales

Chapitre 2 : Navigation dans un environnement

3. Modélisation de l'interaction robot-surface

Un robot mobile au sol interagit avec la surface sous-jacente pour se propulser. Pour remplir cette fonction, il existe de nombreux actionneurs de locomotion différents, tels que des roues, des chenilles, des pattes et même des roues omnidirectionnelles. La figure 2.2 illustre trois exemples réels de robots mobiles terrestres utilisant différentes configurations d'actionneurs. Ces actionneurs, ainsi que les articulations qui les relient au corps du robot, déterminent la structure cinématique et le comportement dynamique du robot. En d'autres termes, ils déterminent la configuration de locomotion du robot. [19]

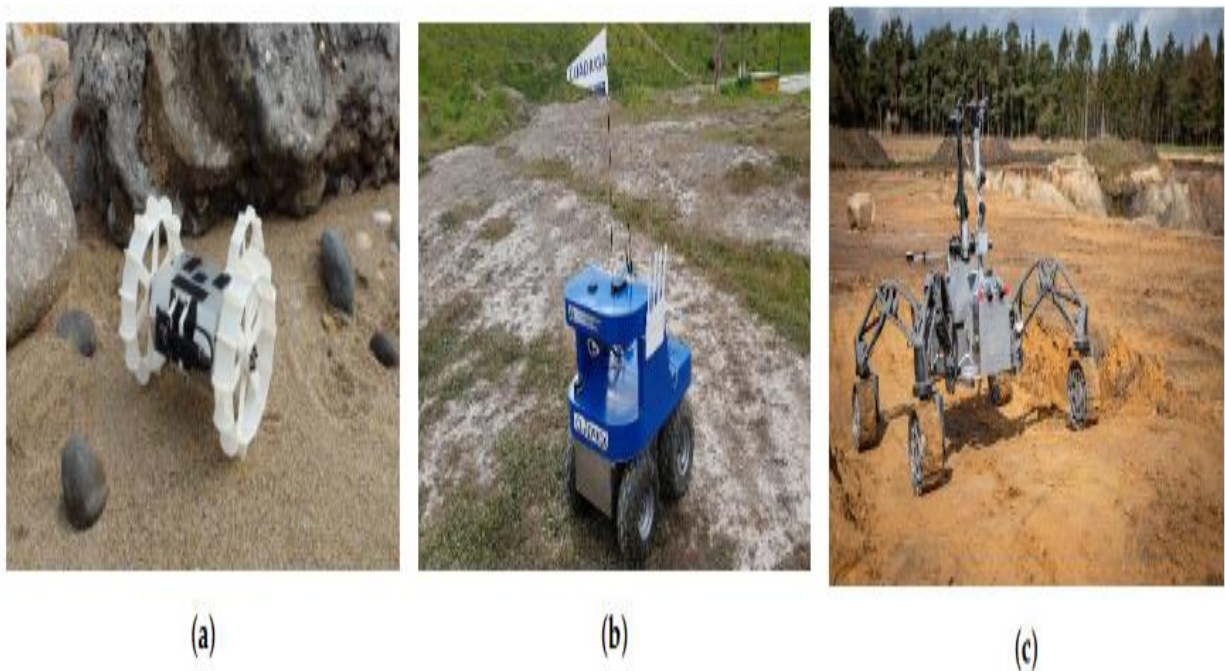


Figure 2. 2 : Exemples de robots mobiles au sol avec différentes configurations cinématiques.

Le rover Koguma (a) a une locomotion à entraînement différentiel. (b) est un robot Skid-steering. SherpaTT (c) a quatre orientables.

Certains d'entre eux impliquent des contraintes liées à la planification de trajectoire, d'autres utilisent une configuration roue sur pattes pour exécuter un mode de locomotion appelé "Wheel-Watling". Ce mode est conçu pour surmonter les terrains meubles dans lesquels un robot pourrait rester coincé. Les algorithmes de planification de trajectoire reconnaissant cette capacité de reconfiguration sont indispensables pour ce type de robot, car ils peuvent trouver des trajectoires qui tirent parti de leur grande adaptabilité. Dans le cas de la planification globale. [20]

Chapitre 2 : Navigation dans un environnement

4 .Représentations topologiques de l'environnement

Il existe plusieurs façons de représenter méthodes de représentation des environnements pour la planification de chemin. Une partie d'entre elles s'intéressent uniquement à la géométrie de l'environnement, alors que d'autres se basent sur des environnements informés pour obtenir des comportements de navigation plus crédibles. [21]

4.1. Représentations approximatives de l'environnement

- Décrire l'espace libre – de navigation – avec des formes géométriques simples telles que des carrés ou des segments.
- Deux modèles entrent dans cette catégorie :
 - les modèles à base de grilles.
 - les cartes de cheminement. [20]

a. Modèles à base de grilles

- Le premier modèle de représentation approximative utilise des grilles régulières
- L'environnement est pavé par des cellules, qui peuvent avoir trois états : libre, partiellement obstruée, et obstacle. [20]

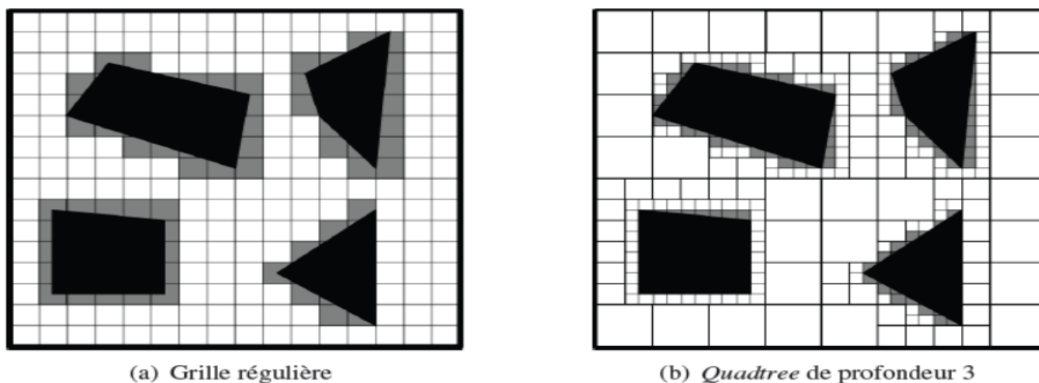


Figure 2. 3 : Modèles à base de grilles

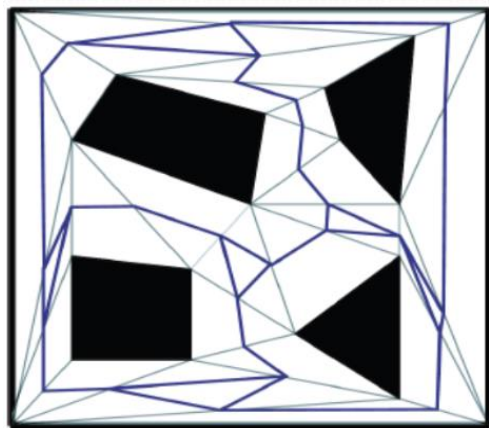
- La précision de la représentation obtenue dépend directement de la taille des cellules utilisées.
 - L'occupation mémoire de cette méthode constitue ainsi son premier point faible.
- Une évolution de ce modèle est apparue sous la forme des grilles hiérarchiques.
 - Décrire l'espace navigable par une succession de grilles de plus en plus précises, organisées sous forme d'arbre.

Chapitre 2 : Navigation dans un environnement

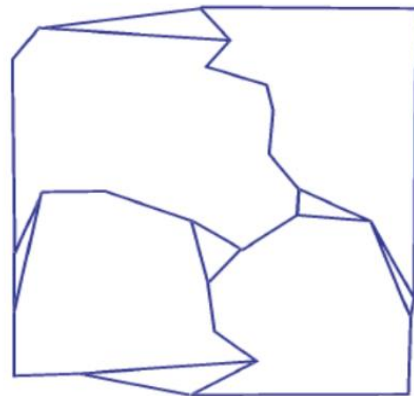
- Fortement utilisée en animation comportementale du fait de leur simplicité de mise en œuvre et de leur rapidité d'exploitation.
- Les grilles sont très difficilement généralisables à des environnements quelconques (des obstacles non alignés sur les axes, un cadre de simulation où l'individu devra se déplacer finement). [20]

b. Cartes de cheminement

- Discrétiser l'espace navigable sous la forme d'un réseau de chemins.
 - Ce réseau est obtenu en reliant des points clefs répartis à l'intérieur de l'environnement.
- Plusieurs méthodes existent pour créer des cartes de cheminement, différentes dans la manière de créer et de relier ces points clefs. [20]



(a) Calculs sur l'environnement d'origine



(b) Exemple de carte de cheminement

Figure 2. 4 : Cartes de cheminement

- Diagramme de Voronoï
 - Basée sur une notion d'équidistance aux obstacles de l'environnement.
 - Un ensemble de sites sont évalués au sein de l'environnement, correspondant aux obstacles, dont les intersections vont former les points clefs.
- La triangulation de Delaunay
- Une évolution de ce modèle est apparue sous la forme des grilles hiérarchiques.
 - Décrire l'espace navigable par une succession de grilles de plus en plus précises, organisées sous forme d'arbre.

Chapitre 2 : Navigation dans un environnement

- Cartes de cheminement probabilistes
 - Les points clefs sont obtenus par une distribution aléatoire au sein de l'environnement navigable.
 - Deux points sont ensuite reliés s'il existe un chemin libre de collision entre eux.
- Fournir une description très condensée de l'environnement, ne nécessitant une prise de décision qu'au niveau des points clefs.
- Leur exploitation devient très difficile lorsqu'il s'agit de gérer le croisement des entités le long d'un même chemin.
- Leur fonctionnement est trop éloigné du comportement humain pour être exploité dans des simulations réalistes. [20]

4.2. Représentations exactes de l'environnement

- Organiser les données spatiales tout en conservant intégralement les informations qu'elles contiennent à l'origine.
- Découper l'espace navigable en cellules convexes de différentes formes.
 - Il existe plusieurs modèles pour effectuer cette subdivision. [20]

Triangulation de Delaunay

Créer un ensemble de triangles en réunissant des points fournis en entrée.

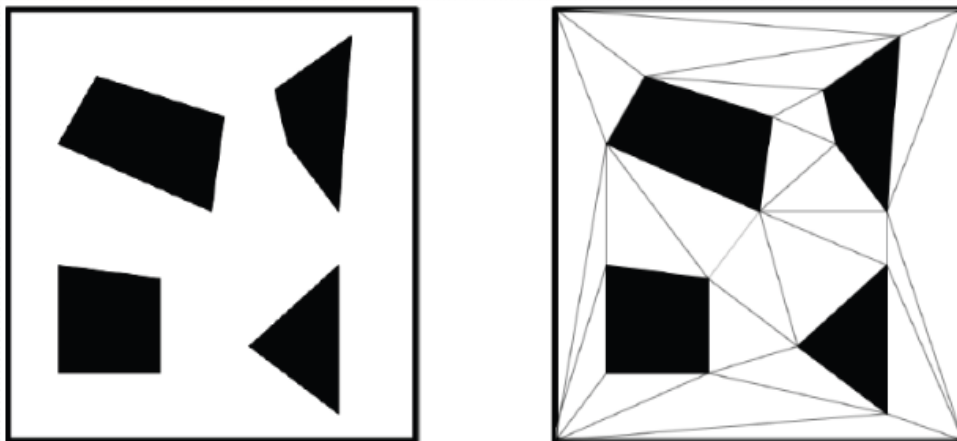


Figure 2. 5 : Triangulation de Delaunay. [20]

Chapitre 2 : Navigation dans un environnement

Ces représentations reproduisent exactement l'environnement d'origine tout en l'organisant de manière plus accessible. Pour des environnements à géométrie complexe, notamment contenant des courbes, le nombre de triangles obtenus augmente très rapidement. [20]

4.3. Le modèle de champs de potentiels

Les champs de potentiels caractérisent les obstacles de l'environnement par des forces de répulsion, et les buts par des forces d'attraction. [20]

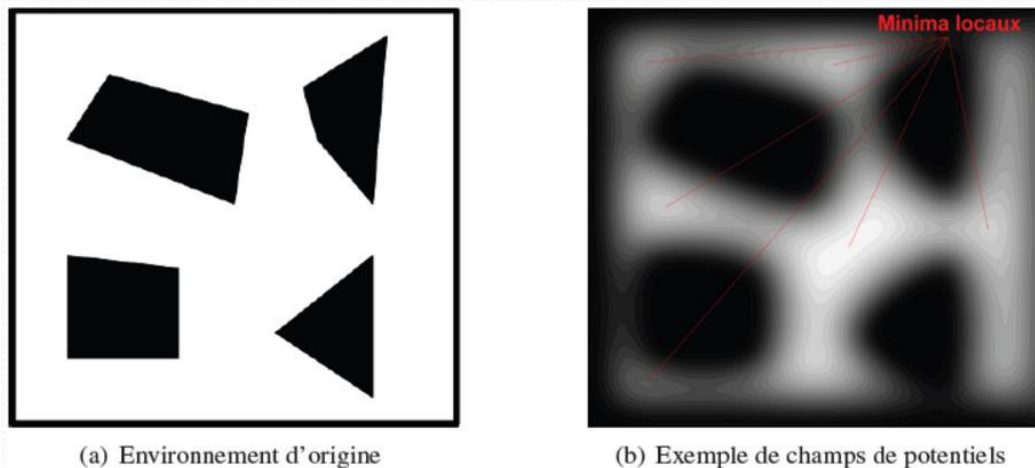


Figure 2. 6 : Le modèle de champs de potentiels

Un gradient de forces, ou champ de potentiel, est déduit en chaque point de l'environnement comme étant une somme pondérée, le plus souvent par la distance, des potentiels de répulsion et du potentiel lié au but. La navigation d'une entité est alors simulée par une descente de gradient depuis sa position dans l'environnement. [20]

Les méthodes basées sur les champs de potentiels s'avèrent simples et efficaces.

Le problème des minima locaux : zones de l'environnement où un potentiel minimal isolé apparaît. Pousser l'entité à se déplacer vers le minimum local le plus proche, qui ne représente pas forcément son but. [20]

5. Planification de chemin

Le procédé de planification de chemin a pour but d'extraire un chemin reliant la position courante de l'entité à une destination donnée.

Cette planification consiste à utiliser des algorithmes de parcours de graphe, ou bien la descente de gradient pour le cas des champs de potentiel. [20]

Chapitre 2 : Navigation dans un environnement

5.1. Les algorithmes de parcours de chemin

- Différents algorithmes de parcours de graphe peuvent être utilisés dans le cadre de la planification de chemin.
- La minimisation de coût est l'un des critères sur lesquels se basent ces algorithmes pour extraire le plus court chemin.
- Ce coût représente souvent une notion de distance spatiale. [20]

5.1.1. L'algorithme de Dijkstra

L'algorithme de Dijkstra (du nom de sa découverte, E.W. Dijkstra) résout le problème de trouver le chemin le plus court entre un point dans un graphique (la source) et une destination. Il s'avère que l'on peut trouver les chemins les plus courts d'une source donnée à tous les points d'un graphique en même temps, d'où ce problème est parfois appelé le problème des chemins les plus courts à source unique. [22]

Étapes de l'algorithme de Dijkstra :

1. Définissez la distance de tous les points sur l'infini, à l'exception de la distance définie du point de départ sur 0.
2. Définissez tous les points, y compris le point de départ en tant que nœud non visité.
3. Définissez le nœud non visité avec la plus petite distance actuelle comme nœud actuel "C".
4. Pour chaque voisin « N » de votre nœud actuel : ajoutez la distance actuelle de « C » avec le poids de l'arête reliant « C » - « N ». Si elle est inférieure à la distance actuelle de « N », définissez-la comme la nouvelle distance actuelle de "N."
5. Marquez le nœud actuel "C" comme visité.
6. Répétez l'étape ci-dessus à partir de l'étape 3 jusqu'à ce que le point de destination soit visité. [23]

Avantages et inconvénients

- Permet de trouver l'ensemble des meilleurs chemins entre deux nœuds du graphe.
- L'intérêt majeur de cet algorithme est sa robustesse.
- Son point faible est sa complexité calculatoire. [20]

Chapitre 2 : Navigation dans un environnement

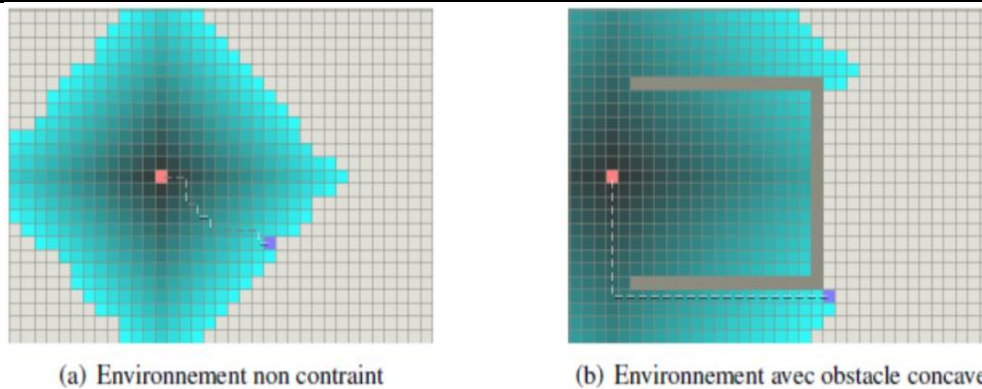


Figure 2. 7 : Un exemple d'algorithme Dijkstra

5.1.2. L'algorithme A*

A-Star (A*) (Hart, Nilsson et Raphael, 1968) est un des algorithmes classiques les plus robustes car il trouve toujours un chemin s'il en existe un. Il allie également un bon ratio performance/qualité du chemin trouvé comparativement aux autres algorithmes, ce qui est nécessaire dans un jeu vidéo. Il faut que le temps de réaction du programme par rapport à l'ordre du joueur, soit proche de l'instantané. Contrairement à la majorité des algorithmes, il utilise une heuristique de la distance entre la position du personnage et le but, c'est une des caractéristiques qui permet de le différencier. Son seul désavantage notable est le fait qu'il ne trouve pas toujours le meilleur chemin si l'heuristique n'est pas admissible alors que l'algorithme de Dijkstra le trouvera toujours.

Pour obtenir une heuristique admissible, il faut que l'heuristique ne surestime dans aucun cas la distance jusqu'à la destination. Atteindre une destination par le chemin le plus court est une activité quotidienne que nous faisons tous. A-star (également appelé A*) est l'un des algorithmes de recherche les plus performants pour trouver le chemin le plus court entre les nœuds ou les graphiques. Il s'agit d'un algorithme de recherche informé, car il utilise des informations sur le coût du chemin et utilise également des heuristiques pour trouver la solution. [24]

A* atteint l'optimalité et l'exhaustivité, deux propriétés précieuses des algorithmes de recherche. Lorsqu'un algorithme de recherche a la propriété d'optimalité, cela signifie qu'il est assuré de trouver la meilleure solution possible. Lorsqu'un algorithme de recherche a la propriété d'être complet, cela signifie que si une solution à un problème donné existe, l'algorithme est assuré de la trouver. [24]

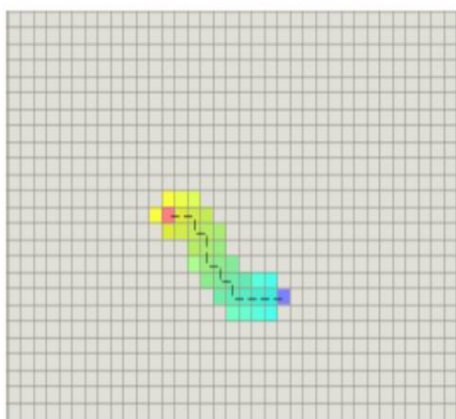
Chapitre 2 : Navigation dans un environnement

Étapes de l'algorithme A* :

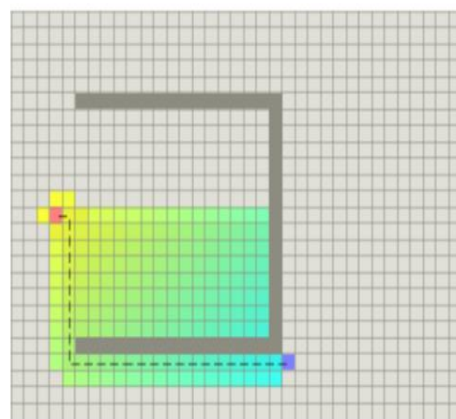
1. Définissez la distance de tous les points sur l'infini, à l'exception de la distance définie du point de départ sur 0.
2. Définissez tous les points, y compris le point de départ en tant que nœud non visité.
3. Définissez le nœud non visité avec la plus petite distance actuelle comme nœud actuel "C".
4. Pour chaque voisin « N » de votre nœud actuel : ajoutez la distance actuelle de « C » avec le poids de l'arête reliant « C » - « N » et le poids au point de destination (heuristique). Si elle est inférieure à la distance actuelle de "N", définissez-la comme la nouvelle distance actuelle de "N".
5. Marquez le nœud actuel "C" comme visité.
6. Répétez l'étape ci-dessus à partir de l'étape 3 jusqu'à ce que l'un des voisins "N" soit le point de destination. [23]

Avantages et inconvénients

- Fonctionne d'une façon similaire à l'algorithme de Dijkstra, mais en ajoutant un coût prédictif aux nœuds correspondant au reste du chemin à parcourir.
- L'avantage de cet algorithme réside dans sa rapidité calculatoire.
- Son inconvénient majeur réside dans le recours à une heuristique. [20]



(a) Environnement non contraint



(b) Environnement avec obstacle concave

Figure 2. 8: Un exemple d'algorithme A*

Chapitre 2 : Navigation dans un environnement

- Critères de planification de chemin
 - Des études en psychologie ont montré que l'information de distance est nécessaire, mais certainement pas suffisante pour rendre compte de la complexité du raisonnement humain.
 - D'autres critères d'évaluation sont proposés
 - La notion de déviation minimale,
 - Le concept de charge cognitive associée à la recherche d'un chemin, avec une stratégie de chemin le plus simple,
 - Des facteurs qualitatifs via des préférences... [20]

Conclusion

Les robots mobiles sont de plus en plus utilisés dans diverses industries et leur capacité à naviguer de manière sûre et efficace dans des environnements dynamiques est cruciale pour leur omniprésence. La complexité de la navigation est due à la dimension temporelle ajoutée dans l'espace de recherche et les interactions subtiles et imprévisibles entre les agents dynamiques tels que les humains, les autres robots ou les véhicules. Pour relever ces défis, les robots mobiles sont équipés de nombreux équipements intelligents et utilisent des techniques de navigation avancées. La planification sûre du chemin est une fonction clé de la navigation et la sélection judicieuse de la technique de navigation est essentielle. Avec des avancées technologiques continues, la navigation des robots mobiles est un domaine en constante évolution qui ouvre la voie à de nouvelles applications et de nouvelles possibilités pour améliorer l'efficacité et la sécurité des robots mobiles dans différents domaines.

Chapitre 03 : Reconstruction 3D de l'environnement

Chapitre 03 : Reconstruction 3D de l'environnement

1. Introduction

Après avoir étudié les types de robots, leurs usages et leur importance, et abordé la modélisation de l'environnement et l'interaction des robots avec celui-ci, et les différentes représentations topologiques de l'environnement et les différents algorithmes de parcours, nous nous concentrerons dans ce chapitre sur les étapes de reconstruction d'un environnement dynamique à l'aide d'un lidar bidimensionnel.

2. Objectif

L'objectif principal du travail est de proposer une alternative plus économique à la technologie lidar 3D, utilisant des composants facilement accessibles, mais permettant tout de même la création d'une représentation précise et dynamique de l'environnement. Nous utilisons du LIDAR 2D et un servo motor « Futaba S3003 » avec Raspberry Pi et Python pour collecter les données de distance et d'angle de l'environnement et générer une carte d'environnement 3D qui suit les changements en temps réel.

3. Conception de notre système

Ce système est conçu pour reconstruire un environnement dynamique à l'aide d'un lidar 2D et d'un servomoteur. Le lidar 2D est connecté au Raspberry Pi pour collecter des données de distance et d'angle pour chaque point de l'environnement. Les étapes comprennent la connexion des composants, l'utilisation des bibliothèques nécessaires, la programmation pour collecter des données à partir de différents angles de rotation du servomoteur, la création d'une carte de l'environnement à l'aide d'algorithmes de cartographie et la reconstruction d'une image 2D ou 3D.

Chapitre 03 : Reconstruction 3D de l'environnement

3.1 Conception générale

Dans cette partie, on décrit l'architecture globale de notre système.

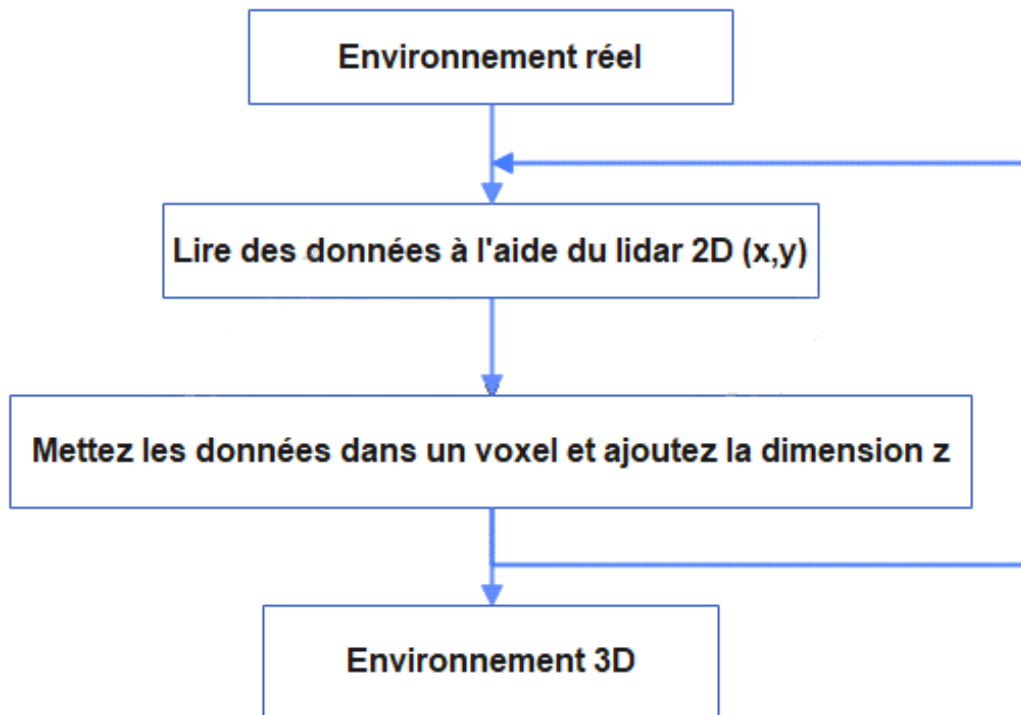


Figure 3. 1: Schéma général du système

L'application commence par placer l'appareil dans un environnement réel. Les données sont lues à partir d'un lidar bidimensionnel sous la forme x y et stockées dans un voxel, puis la dimension z est ajoutée dans un voxel. Une fois les valeurs de voxel mises à jour en fonction des lectures du capteur, le résultat est affiché en créant un graphique 3D représentant les voxels, où les voxels sont tracés sur le graphique 3D et colorés dans des couleurs aléatoires. Lorsque nous exécutons le code, une fenêtre apparaît contenant le contour 3D représentant le voxel mis à jour. Les obstacles détectés apparaîtront sous forme de points colorés dans l'espace 3D, et chaque obstacle peut avoir une couleur différente. Ainsi, nous obtenons un environnement.

3.2 Conception détaillée

Dans cette partie, on décrit l'architecture détaillée de chaque processus de notre système.

Chapitre 03 : Reconstruction 3D de l'environnement

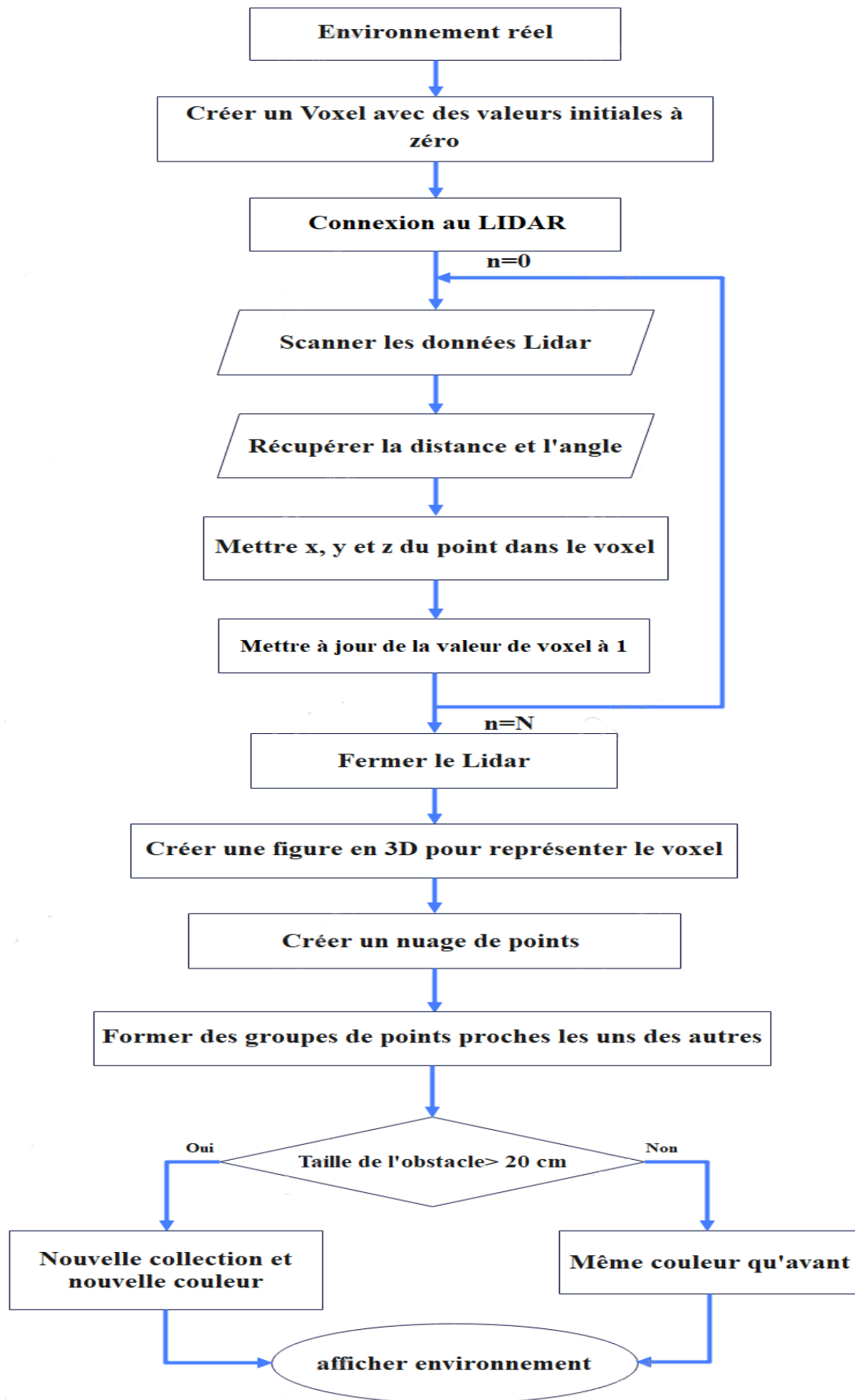


Figure 3. 2: Organigramme de système.

Chapitre 03 : Reconstruction 3D de l'environnement

3.2.1 Explication de l'organigramme d'un système

Un voxel de la taille spécifiée est d'abord créé avec des valeurs initialisées à zéro.

- **Un voxel** : est une unité tridimensionnelle dans le domaine des données 3D, c'est un cube tridimensionnel contenant une valeur ou une information. Un voxel est similaire à l'idée utilisée dans une image pixel, où un pixel est une unité de base d'une image bidimensionnelle. Il peut être utilisé pour représenter une distance 3D (telle que des points dans un espace 3D) ou pour représenter des propriétés spécifiques (telles que la densité ou la variation de couleur) dans des données 3D.

Après cela, ouvrez l'appareil Lidar et démarrez le moteur

Le voxel est mis à jour en fonction des lectures reçues du capteur. Les coordonnées de chaque point sont calculées en fonction de la distance et de l'angle détectés par le capteur, puis converties en coordonnées de voxel 3D.

1. Les lectures sont obtenues à partir du lidar à chaque cycle.
2. A chaque lecture, la distance et l'angle sont extraits.
3. L'angle vertical est calculé en utilisant l'angle et la distance.
4. Les coordonnées 3D du voxel (x, y, z) sont calculées en utilisant la distance et l'angle. Où les coordonnées polaires détectées par le capteur sont converties en coordonnées de voxel 3D, et la distance et l'angle sont convertis en trois axes de voxel (x, y, z) en fonction de la taille du voxel et des variables `size_x`, `size_y` et `size_z`.

La valeur de voxel des points détectés LIDAR est mise à jour et convertie en coordonnées de voxel 3D.

Après avoir calculé les coordonnées de chaque point (x, y, z) en fonction de la distance et de l'angle détectés par le capteur.

Pour la dimension z, elle est calculée avec servo motor sur chaque cycle.

Mettre à jour la valeur du voxel dans la cellule correspondant à ces coordonnées avec une valeur 1

De cette manière, la valeur du voxel dans les positions détectées du capteur est mise à jour avec une valeur de un, et le reste des cellules du voxel restent à leur valeur par défaut (zéro), permettant la représentation de la présence ou de l'absence d'obstacles dans l'espace tridimensionnel représenté par le voxel.

Chapitre 03 : Reconstruction 3D de l'environnement

Après avoir terminé les lectures, arrêtez le moteur et déconnectez le capteur lidar.

Ensuite, nous créons une forme 3D pour représenter le voxel mis à jour sous forme de nuage de points.

Nous divisons les points en groupes. Si la distance entre les points est supérieure à 20 cm, un nouveau groupe sera formé dans une couleur différente. Nous le placerons dans une forme géométrique tridimensionnelle. Si la distance est inférieure à 20 cm, il prend la même couleur que le précédent.

Afficher les obstacles détectés sous forme de points colorés dans l'espace 3D, chaque obstacle peut avoir une couleur différente.

A) Scanner les données

Au début du travail, nous plaçons l'appareil lidar dans un environnement réel pour qu'il puisse scanner les données, comme illustré.

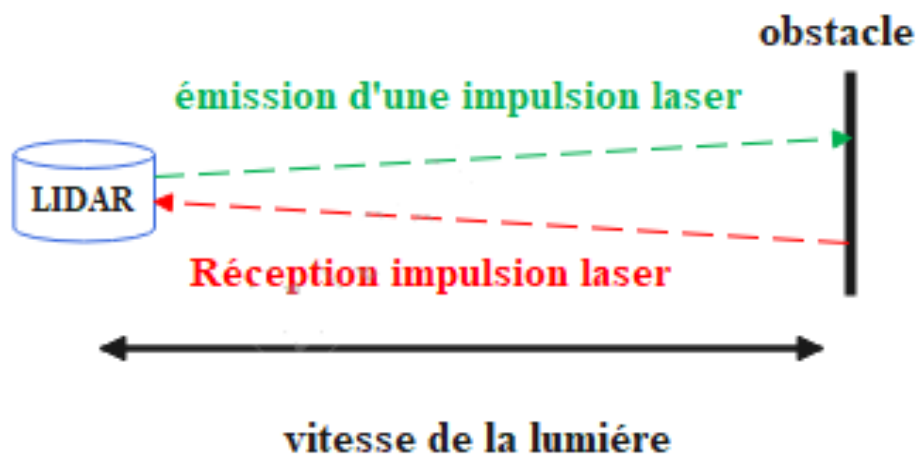


Figure 3. 3 : Le lidar envoie des signaux et les reçoit.

Émission du signal : Le lidar envoie des signaux lumineux très courts, tels que des lasers, dirigés vers l'environnement qui l'entoure.

Réception du signal : Le signal réfléchi par les différents obstacles de l'environnement, tels que les objets et les obstacles, est reçu. Le lidar utilise un récepteur pour recevoir ces signaux réfléchis.

Chapitre 03 : Reconstruction 3D de l'environnement

Mesure de la distance : En fonction du temps que met le signal pour aller de l'émetteur à l'obstacle et revenir, la distance peut être calculée avec précision. Cela repose sur une mesure précise du temps et l'utilisation des équations de vitesse et de temps.

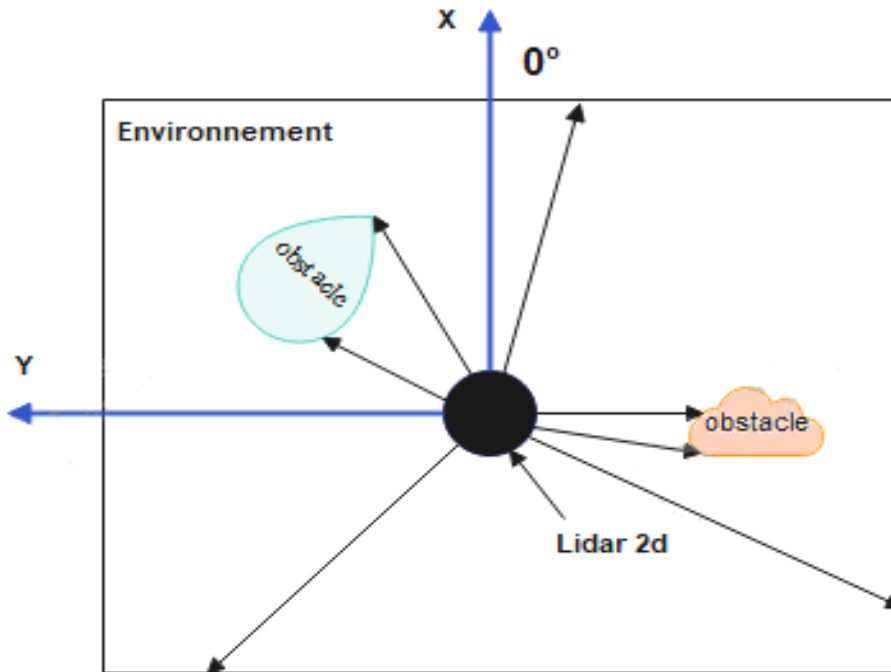


Figure 3. 4 : Système de numérisation Lidar.

B) Angle de rotation du lidar et détection d'obstacles

L'angle de rotation du lidar dans la pièce et la détection des obstacles sont déterminés par la programmation et les réglages du lidar. L'angle peut être différent selon le modèle utilisé et les exigences de l'application. Le lidar est réglé pour tourner à 360 degrés pour couvrir uniformément toutes les directions de la pièce. Le lidar envoie des rayons dans toutes les directions et reçoit des signaux réfléchis pour détecter et localiser les obstacles dans l'environnement.

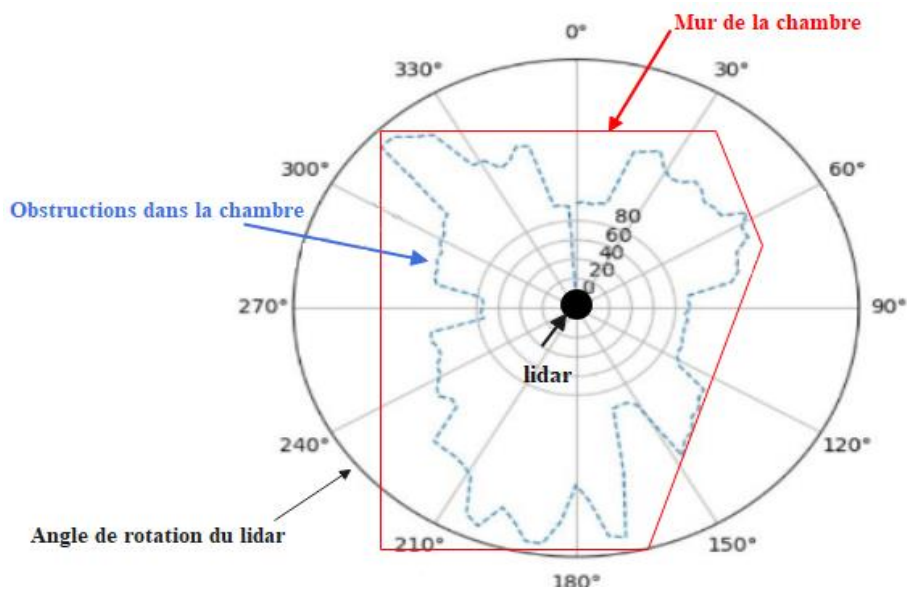


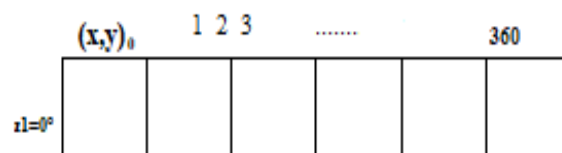
Figure 3. 5 : Lecture des obstacles par LIDAR à l'intérieur de la pièce

Lorsque le lidar est installé dans la pièce, il tourne à un angle de 360 degrés, envoyant des rayons dans une large gamme d'angles de pièce pour couvrir toutes les directions.

C) Structure de données d'un voxel

Collecte de données

Les coordonnées (x, y) sont enregistrées dans les voxel, qui sont des unités de stockage tridimensionnelles. La valeur initiale de tous les voxel est définie à $(x_0, y_0, z_0) = 0$, puis la valeur de z est fixée. Le lidar effectue ensuite le processus de numérisation et attribue des valeurs aux coordonnées (x, y) . Si le signal est réfléchi, la valeur enregistrée dans le voxel correspondant est 1, sinon elle est enregistrée comme 0.



Une fois que 360 paires de coordonnées (x, y) ont été enregistrées, le servo-moteur change l'angle de 0 à 20 degrés. Cela modifie la valeur de z . Ensuite, le processus de numérisation est répété pour un autre cycle complet, en enregistrant les nouvelles coordonnées (x, y) dans les voxel.

Ce processus est répété 3 fois, conformément à ce qui est indiqué la description donnée.

Chapitre 03 : Reconstruction 3D de l'environnement

(x,y)	1	2	3	360
$\alpha=0^\circ$					
$\alpha=10^\circ$					
$\alpha=60^\circ$					

Figure 3. 6 : Structure de données d'un voxel

Explication de l'algorithme de collecte de données :

1. Nous définissons les variables requises (angle, rotation, x, y, réflexion).
2. Ensuite, nous initialisons le voxel avec une valeur nulle (0).
3. La valeur z et l'angle de démarrage du servomoteur sont réglés sur 0° .
4. Temps de démarrage de la boucle (CYCLES)
5. Démarrage d'une boucle sur la dimension x
6. Démarrage d'une boucle sur la dimension y
7. Utilisez le capteur (lidar) pour obtenir la valeur de réflectivité (réflectance).
8. Stocke la valeur de réflexion dans le voxel approprié.
9. La fin de la boucle intérieure de la dimension y.
10. La fin de la boucle intérieure de la dimension x.
11. Mise à jour de l'angle du moteur pour l'interface d'angle (ANGLE).
12. Mise à jour de la liste basée sur le nouvel angle du servomoteur.
13. Le cycle de cours se termine.
14. Afficher les résultats en voxel

Algorithme voxel

Variables :

M : tableau [3, 360] entier ;

(x, y), z : entier ;

Débu

Pour i <-1 à 3 faire

Pour j <-1 à 360 faire

Lire (M [z, (x, y)]) ;

Ecrire (M [z, (x, y)]) ;

FinPour ;

FinPour ;

Fin.

Figure 3. 7 : Algorithme de voxel

D) Regroupement des points en groupes

Après avoir créé le voxel et l'avoir rempli avec les informations d'environnement, nous collectons maintenant toutes les valeurs convergentes dans la distance, c'est-à-dire si la distance entre deux obstacles est inférieure à 20 cm par exemple (la longueur du support d'appareil), nous les formons comme une seule barrière et mettons une couleur spéciale pour ce nouvel obstacle.

- Si la distance entre deux obstacles est inférieure à 20 cm, il prend la même couleur que le précédent.
- Si la distance entre deux obstacles est supérieure à 20 cm, la maille suivante prend une nouvelle couleur.
- Vous ne pouvez pas répéter la même couleur verticalement.

Comme montré :

Chapitre 03 : Reconstruction 3D de l'environnement

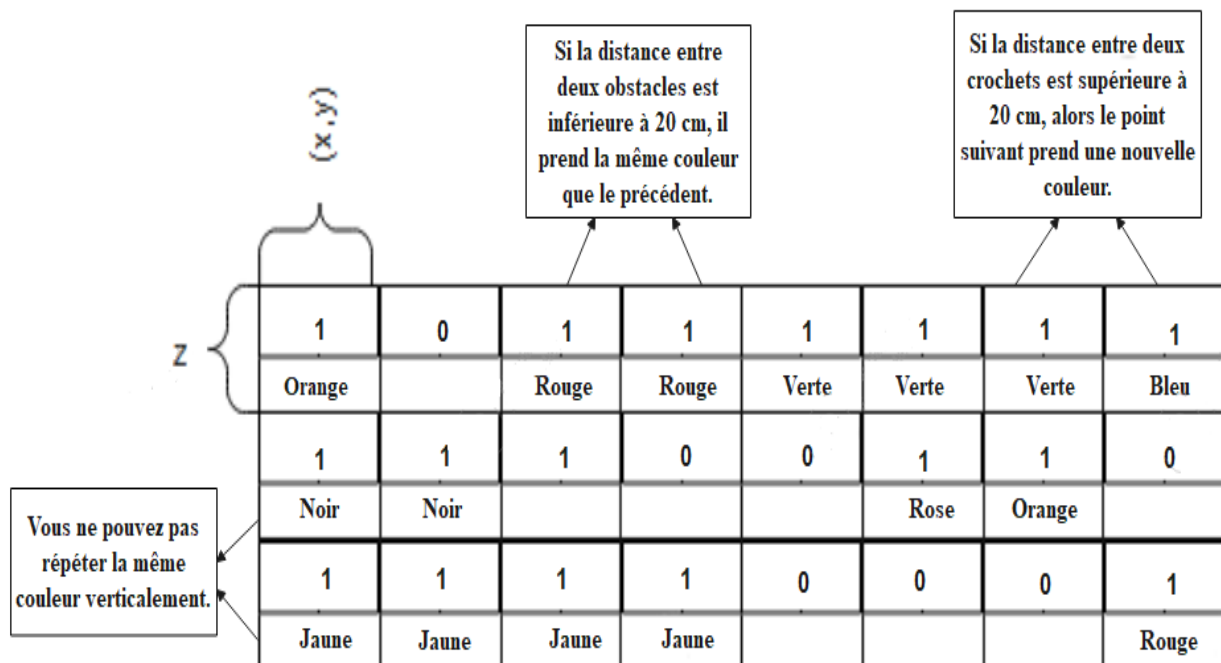


Tableau 2 : Mettez les obstacles en groupes

Conclusion

Dans ce chapitre, nous avons présenté les spécifications pour la conception de notre système, à la fois d'un point de vue général et détaillé, ce qui constitue l'une des étapes les plus importantes du processus de développement du système et de compréhension de sa formation.

Dans la première étape, nous avons décrit les étapes du système de manière générale et sans entrer dans les détails. Dans la deuxième étape, nous avons décrit la configuration détaillée du système et nous avons étayé la description à l'aide d'un diagramme de flux du système.

Dans le prochain chapitre, nous illustrerons la réalisation de notre système en présentant certains des résultats obtenus, ainsi que les outils et logiciels utilisés pour mener à bien ce travail.

Chapitre 04 : Implémentation et résultats

Chapitre 04 : Implémentation et résultats

1. Introduction

Après avoir établi la conception de notre système d'un point de vue général et détaillé qui est à la base de notre système en général, ce chapitre donne un aperçu du processus suivi pour atteindre notre objectif. Nous allons d'abord présenter les moyens utilisés puis présenter les résultats obtenus.

2. Configuration matérielle

Raspberry

Le Raspberry Pi est une série de petits ordinateurs à carte unique développée au Royaume-Uni par la fondation Raspberry Pi dans le but de promouvoir l'enseignement des sciences informatiques de base dans les écoles et dans les pays en développement. Plusieurs générations de Raspberry Pi ont été publiées.

Tous les modèles sont équipés d'un système sur puce (SoC) Broadcom avec une unité centrale de traitement (CPU) compatible ARM intégrée et une unité de traitement graphique (GPU) intégrée.

La vitesse du processeur varie de 700 MHz à 1,4 GHz pour le modèle Pi 3 B+ ; la mémoire embarquée varie de 256 Mo à 1 Go de RAM. Des cartes Secure Digital (SD) au format Micro Secure Digital High Capacity (SDHC sur les premiers modèles) sont utilisées pour stocker le système d'exploitation et la mémoire du programme.

Les cartes disposent d'un à quatre ports Universal Serial Bus (USB). Pour la sortie vidéo, le HDMI et la vidéo composite sont pris en charge, avec une prise jack standard de 3,5 mm pour la sortie audio. Une sortie de niveau inférieur est fournie par plusieurs broches d'entrée/sortie à usage général (GPIO), qui prennent en charge des protocoles courants tels qu'I²C. Les modèles B disposent d'un port Ethernet 8P8C et les modèles Pi 3 et Pi Zéro W ont une connectivité Wi-Fi 802.11n et Bluetooth intégrée. Les prix varient de 5 à 35 dollars américains. [25] Le type de framboise utilisé dans ce travail est Raspberry pi 3.

Chapitre 04 : Implémentation et résultats



Figure 4. 1 : Raspberry Pi 3B+

Servo motor

Un servomoteur est un variateur de vitesse et il est utilisé dans de nombreuses productions industrielles ainsi que pour l'automatisation des processus. Ces moteurs sont principalement utilisés pour les applications de contrôle de mouvement, les servomoteurs sont utilisés dans les applications de contrôle de mouvement car ils ont un positionnement de haute précision, une inversion rapide et de grandes performances. Un servomoteur peut être un moteur à courant alternatif ou à courant continu avec un dispositif de détection de position. Fondamentalement, un servomoteur est un dispositif électromécanique dans lequel une entrée électrique va déterminer la position de l'induit du moteur.

Le servo est un appareil compact et son arbre de sortie est positionné d'une certaine manière à l'aide d'un signal servo-codé. Ainsi, le servo maintiendrait la position angulaire jusqu'à ce que le signal codé existe sur la ligne d'entrée. Donc, si le signal codé change, la position angulaire de l'arbre changera. Le servomoteur est contrôlé par un signal et on peut l'appeler un modulateur de largeur d'impulsion. [26]



Figure 4. 2 : Servomoteur Futaba S3003

Chapitre 04 : Implémentation et résultats

Un adaptateur d'alimentation : utilisé pour connecter le Raspberry Pi à l'électricité.

Carte micros 8 GB : utilisée pour stocker le système d'exploitation et les données sur le Raspberry Pi.

Un câble HDMI : utilisé pour connecter le Raspberry Pi à l'écran.

Un clavier et une souris (ou un clavier et une souris sans fil) : utilisés pour interagir avec le Raspberry Pi.

3. Environnements et outils

3.1. L'environnement de développement

3.1.1. Raspbian

Raspbian est un système d'exploitation libre et open source basé sur la distribution Linux Debian, spécialement conçu pour le Raspberry Pi. Sorti en juillet 2012. Il est le système d'exploitation recommandé par la fondation Raspberry Pi pour les projets utilisant un Raspberry Pi. [27]

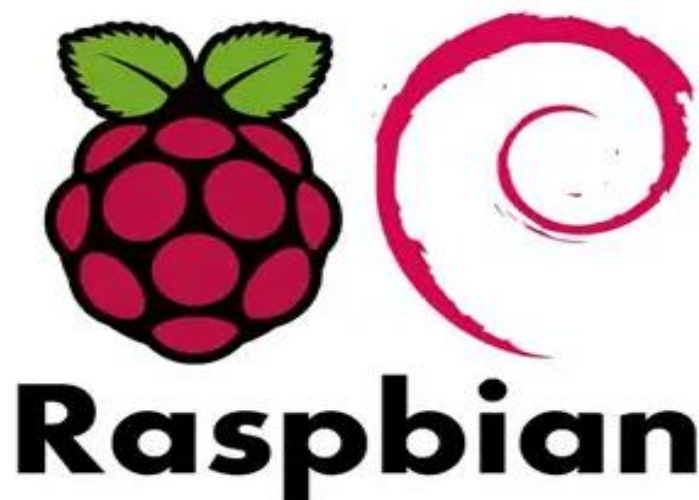


Figure 4. 3 : Logo Raspbian

3.1.2. Thonny

Thonny est un environnement de développement Python et facile à utiliser pour le Raspberry Pi. Il est disponible gratuitement et peut être installé à partir du gestionnaire de paquets standard de Linux, Il est adapté aux débutants comme aux utilisateurs avancés. Grâce à ses fonctionnalités avancées : la syntaxe, débogage et de gestion de packages. [27]

Chapitre04 : Implémentation et résultats

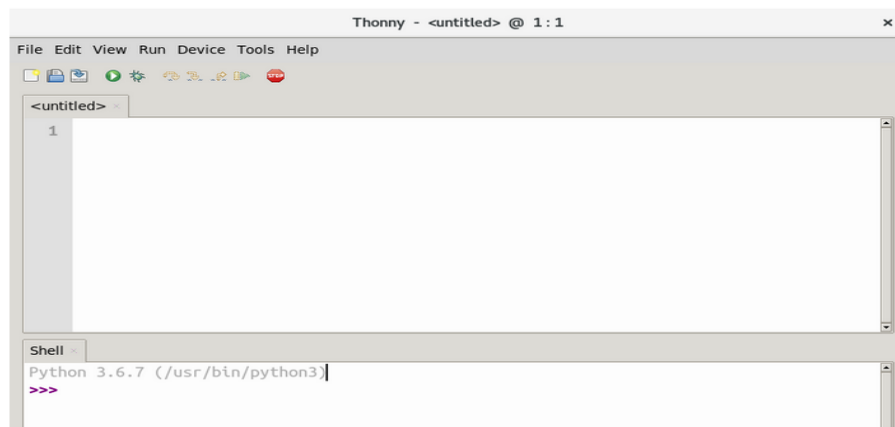


Figure 4. 4 : L'interface de Thonny

3.2. Langage de développement

3.2.1. Python

Python est un langage de programmation puissant, facile à utiliser, facile à lire et à écrire et, avec Raspberry Pi, vous permet de connecter votre projet au monde réel. La syntaxe Python est propre, avec un accent sur la lisibilité et utilise des mots clés anglais standards.

Afin de mettre en œuvre notre travail, nous avons utilisé le langage de programmation Python qui est un langage de programmation de haut niveau interprété et orienté objet avec une sémantique dynamique. Ce langage est très sollicité par une large communauté de développeurs et de programmeurs :

- Python est un langage simple et facile à apprendre.
- Il permet une bonne réduction du coût de la maintenance des codes.
- Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes.
- Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

3.2.2. Les bibliothèques

PyLidar

PyLidar est une bibliothèque Python qui facilite l'écriture de code de traitement lidar en Python. Qui donne accès aux capteurs LIDAR (Light Detection and Ranging). Les capteurs LiDAR utilisent la lumière laser pour mesurer les distances et créer des cartes 3D de leur

Chapitre 04 : Implémentation et résultats

environnement, et sont couramment utilisés dans des applications telles que les véhicules autonomes et la robotique. [28]

RPi.GPIO

RPi.GPIO est une bibliothèque dotée d'un ensemble impressionnant de fonctionnalités : toutes les broches GPIO du Raspberry Pi peuvent être lues, écrites, attachées à des gestionnaires d'interruption et émettre des signaux PWM en même temps. Vous permettant de contrôler des périphériques externes tels que des moteurs et des capteurs. Il est écrit en C et Python. [29]

Matplotlib

Matplotlib est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python. Matplotlib facilite les tâches simples et rend possibles les tâches difficiles en fournissant une large gamme d'outils nécessaires pour créer des graphiques dans de nombreux domaines tels que la science, l'ingénierie, les statistiques, etc. "mpl_toolkits.mplot3d" La bibliothèque aide les utilisateurs à créer divers tableaux et graphiques, notamment des graphiques 2D et 3D, des graphiques linéaires, des graphiques à barres, etc, qui permet de créer des graphiques en 3D à l'aide de Matplotlib. Cette bibliothèque fournit des fonctions pour afficher des graphiques en 3D et créer des graphiques en 3D en utilisant une grande variété de styles différents. [30]

4. Implémentation

a) La première partie : LIDAR 2D

Dans la première phase, nous avons travaillé sur le code LIDAR pour représenter un environnement initial en deux dimensions en fonction des coordonnées LIDAR. Nous avons converti les dimensions x et y en voxel.

Chapitre 04 : Implémentation et résultats

Code LIDAR

```
PORT_NAME = '/dev/ttyUSB0'
DMAX = 4000
IMIN = 0
IMAX = 500

# Define voxel size
voxel_size = 10 # mm
size_x, size_y, size_z = int(DMAX/voxel_size), int(DMAX/voxel_size),
int(DMAX/voxel_size)

# Create voxel
voxel = np.zeros((size_x, size_y, size_z))

def update_voxel():
    #lidar = RPLidar(PORT_NAME)
    lidar = RPLidar(PORT_NAME, baudrate=115200, timeout=10)
    k=0

    lidar.start_motor()

    # Use try-except block to handle Lidar errors and close the device
    properly
    try:
        for scan in lidar.iter_scans(4000):
            for i in range (20):
                for meas in scan:
                    distance = meas[2]
                    theta = np.radians(meas[1])
                    if distance > 0:
                        phi = np.arcsin(np.clip(distance/DMAX, -1, 1))
                        if not np.isnan(phi):
                            x = int(distance * np.cos(phi) * np.cos(theta) /
voxel_size + size_x/2)
                            y = int(distance * np.cos(phi) * np.sin(theta) /
voxel_size + size_y/2)
                            z = 100 +k
                            if x >= 0 and x < size_x and y >= 0 and y < size_y
and z >= 0 and z < size_z:
                                voxel[x, y, z] = 1

                            k=k+1

    except Exception as e:
        print(e)

    finally:
        lidar.stop_motor()
        lidar.disconnect()
```

Chapitre 04 : Implémentation et résultats

```
def main():
    update_voxel()

    # Create figure
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    ax.set_xlim(0, size_x)
    ax.set_ylim(0, size_y)
    ax.set_zlim(0, size_z)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    ax.set_title('Lidar Voxel')

    # Label the connected components
    labeled_voxel = measure.label(voxel)

    # Set different colors for each connected component
    colors = np.random.rand(labeled_voxel.max() + 1, 3)

    # Create voxel data with different shapes and sizes for each connected
    component
    x, y, z = voxel.nonzero()
    ax.scatter(x, y, z, c=colors[labeled_voxel[voxel.nonzero()]], marker='^',
s=10)

    # Show plot
    plt.show()

if __name__ == '__main__':
    main()
```

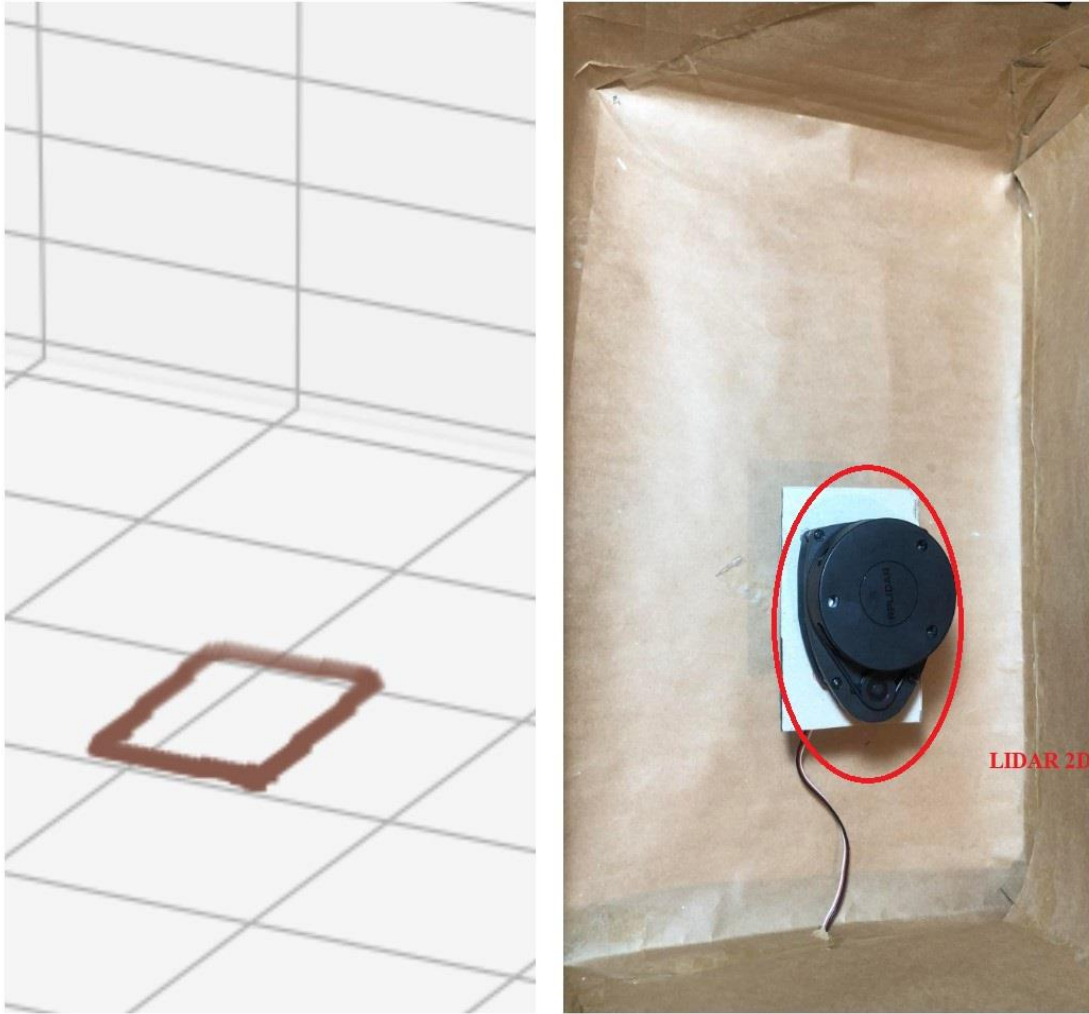


Figure 4. 5 : Image montre un environnement réel et l'environnement obtenu avec le lidar 2D

b) Deuxième partie : servo motor

Nous déterminons la hauteur (dimension z) en utilisant un motor servo, où nous définissons un angle maximal de 60 degrés. Toutes les 20 degrés, nous enregistrons les informations LIDAR pour obtenir un environnement en trois dimensions.

Chapitre 04 : Implémentation et résultats

Code servo motor

```
PORT_NAME = '/dev/ttyUSB0'
lidar = RPLidar(PORT_NAME)
# used to scale data to fit on the screen
max_distance = 0
scan1=[]
def process_data(data):
    for i in range(0,360):
        posXd = 500+(data[i] *cos(radians(i)))
        posYd = 500+(data[i] *sin(radians(i)))
        print("posx:",posXd,",    posy:",posYd,)
        print(i)
        # Do something useful with the data
        pass
scan_data = [0]*360
i=0
tim=0
pas=45
pwm.ChangeDutyCycle(0)
try:
    print(lidar.get_info())
    for scan in lidar.iter_scans():
        #print(scan)
        # print("\n")
        for (_, angle, distance) in scan:
            scan_data[min([359, floor(angle)])] = distance
        tim=tim+1
        if tim % 3==0 :
            sig=(pas/18)+2
            pwm.ChangeDutyCycle(sig)
            scan1.append(scan_data)
            # print(scan1)
            scan_data = [0]*360
            pas=pas+1
            if pas ==120:
                pwm.ChangeDutyCycle(0)
            if tim>=120:
                print(scan1)
                print(len(scan1))
                #RPLidar.clear_input()
            # process_data(scan_data)
except KeyboardInterrupt:
    print('Stopping.')
```

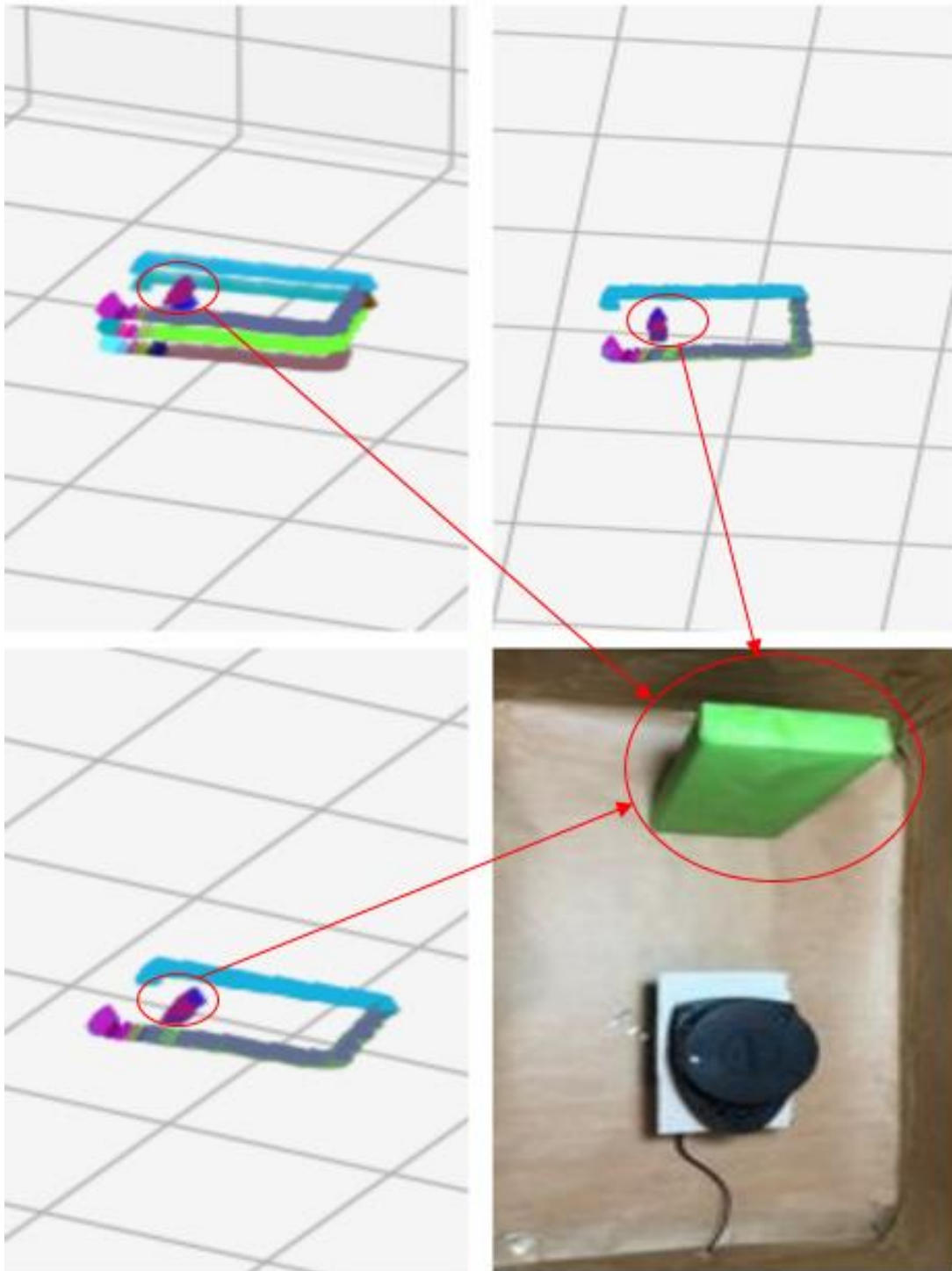


Figure 4. 6 : L'image montre l'environnement réel où un obstacle est placé directement devant le lidar, et l'environnement obtenu le compense de différents côtés.

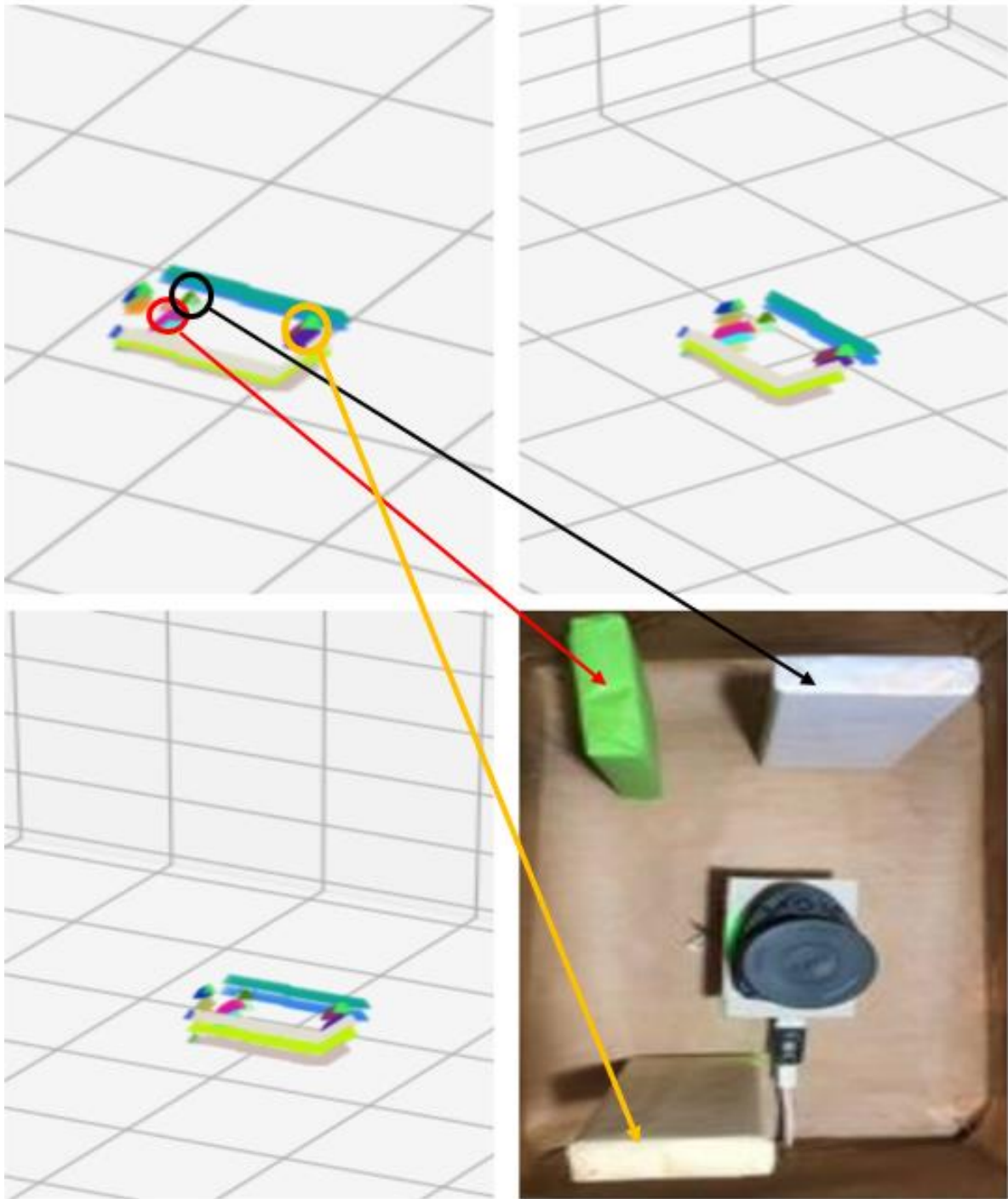


Figure 4. 7 : L'image montre l'environnement réel dans lequel trois obstacles sont placés devant, derrière et à côté du LIDAR, et l'environnement obtenu de différents côtés.

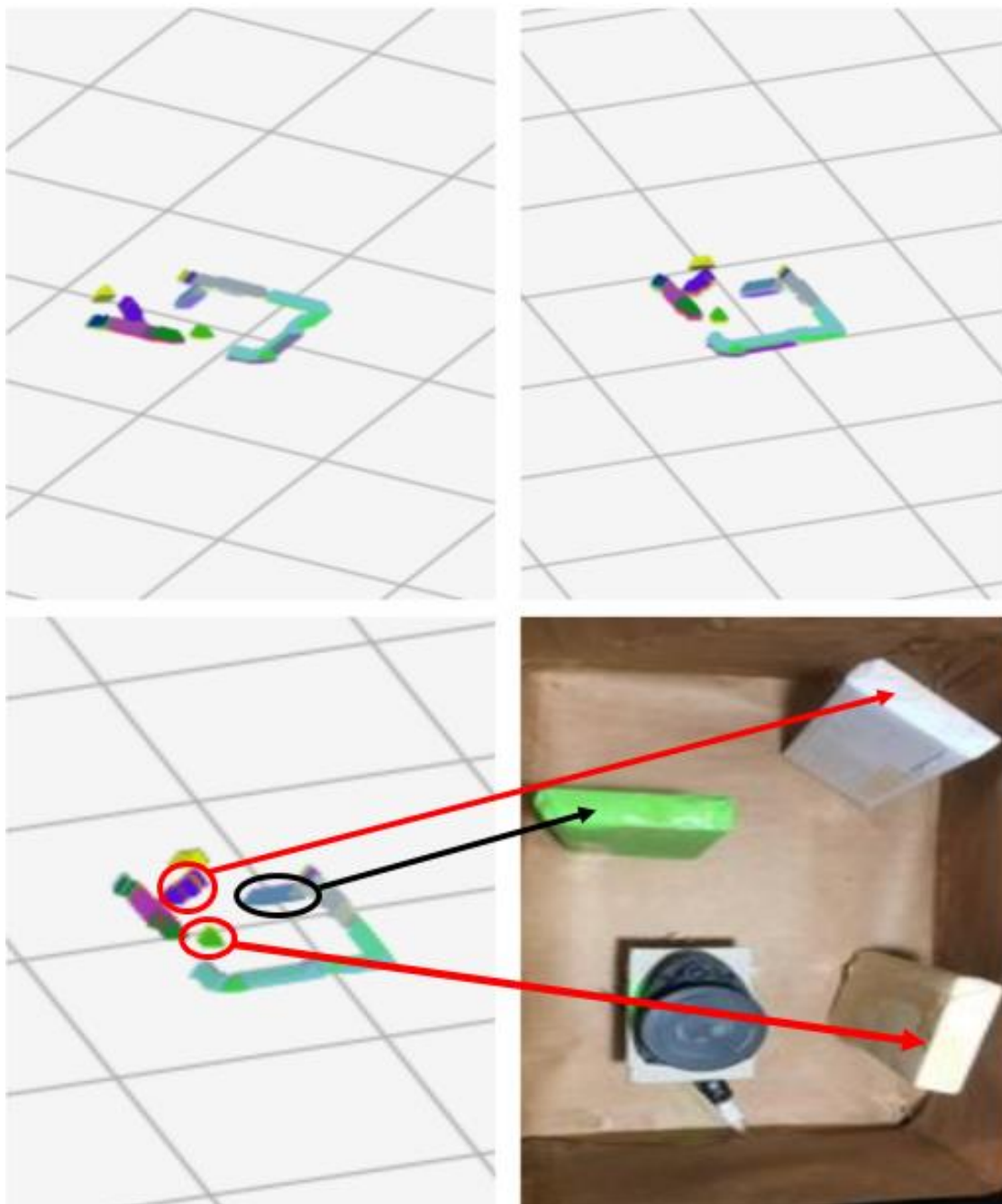


Figure 4. 8 : Image montrant un environnement réel dans lequel trois obstacles sont placés, nous avons changé leurs emplacements et l'environnement obtenu de différents côtés

Chapitre04 : Implémentation et résultats

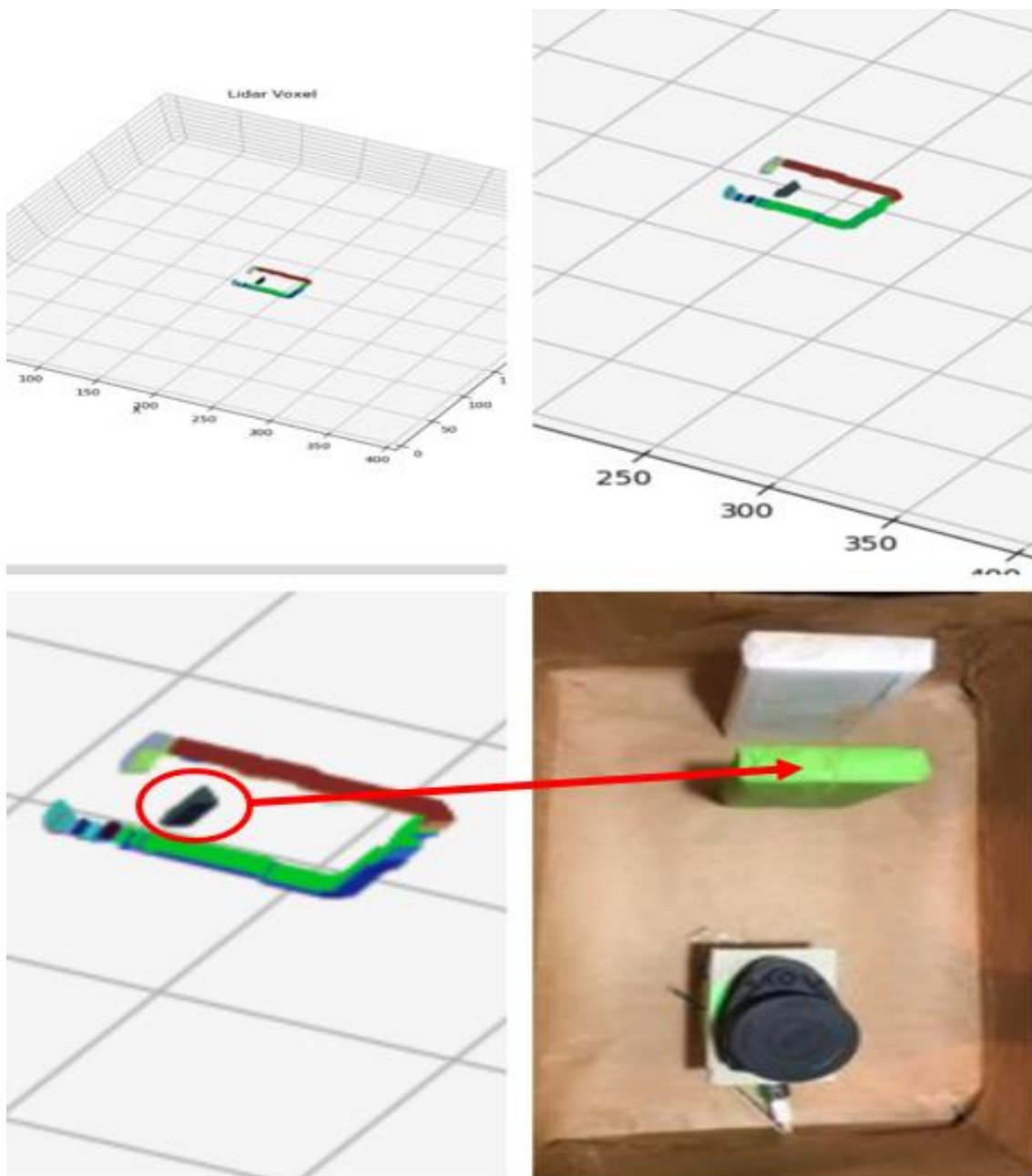


Figure 4. 9 L'image montre un environnement réel dans lequel nous plaçons deux obstacles directement devant le lidar et l'environnement est obtenu sous différents angles.

Chapitre04 : Implémentation et résultats

c) La troisième partie : Support d'appareil

À ce stade, après avoir terminé le code, nous avons fabriqué une mallette de transport matérielle pour placer le lidar et les servomoteurs (2 servomoteurs sont nécessaires).

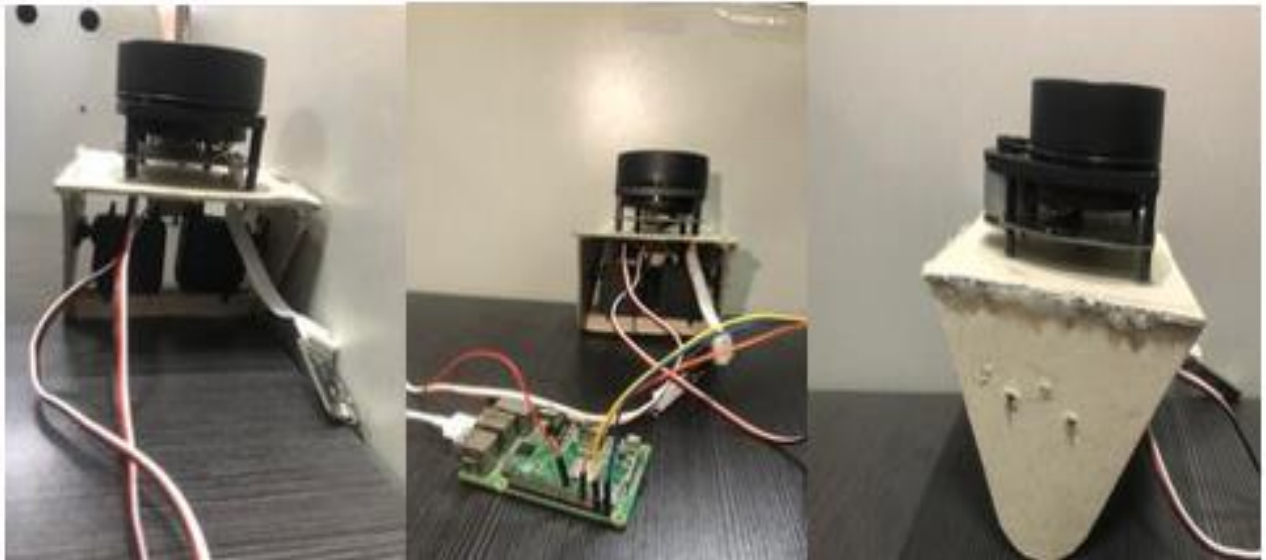


Figure 4. 10 : Support d'appareil

Conclusion

Dans ce chapitre, nous avons décrit la réalisation du système en mettant l'accent sur son exécution détaillée. Nous avons expliqué comment concevoir et développer le système en fonction des exigences spécifiques. La description comprend également des détails techniques sur l'exécution du système, y compris les outils utilisés dans le processus d'exécution. Les outils spécifiques utilisés ont été mentionnés et expliqués quant à leur utilisation dans la mise en œuvre du travail. De plus, il est mentionné que des résultats ont été obtenus à la fin, et ces résultats ont été présentés et expliqués.

Conclusion générale

Conclusion générale

Dans ce mémoire, nous avons reconstruit un environnement réel en utilisant un lidar 2D et un servo motor. Une boîte a été fabriquée pour abriter le lidar et le servo motor, assurant ainsi leur mouvement synchronisé. Lorsque le lidar tourne, le servo motor ajuste l'angle du lidar pour obtenir la dimension "Z" qui représente la hauteur.

De cette manière, nous obtenons les coordonnées tridimensionnelles (X, Y, Z) de l'environnement, qui sont ensuite affichées dans un logiciel en trois dimensions Voxel. Les coordonnées horizontales (X, Y) sont obtenues à partir du lidar, tandis que les coordonnées verticales (Z) sont obtenues à partir du servo motor.

Limites et lacunes

Parmi les principales lacunes de ce projet :

Efficacité : Il peut y avoir un défi pour atteindre une efficacité élevée dans le processus d'agrégation des points et leur conversion en obstacles tridimensionnels, en particulier lorsque de nombreux points sont regroupés.

Précision : Nous pouvons faire face à des défis pour garantir une formation précise des obstacles, notamment en cas de chevauchement ou d'interférence entre les points.

Gestion des dimensions différentes : L'agrégation des points et leur conversion en obstacles tridimensionnels peuvent être difficiles lorsque les points sont de dimensions et de formes différentes, ce qui peut entraîner une distorsion ou une irrégularité des obstacles résultants.

Coût et complexité : La mise en œuvre de ce projet peut nécessiter l'utilisation de techniques et d'outils avancés, ce qui augmente le coût et la complexité générale du projet.

De plus, la boîte contenant les appareils peut ne pas être pratique dans la forme souhaitée, car sa fabrication doit être précise pour obtenir les résultats souhaités.

Travail futur

Il existe de nombreuses améliorations qui peuvent être explorées dans les projets futurs. L'une de ces améliorations concerne l'agrégation des points, où lorsque la distance entre ces points est inférieure à la taille de l'appareil ou de la boîte qui contient les dispositifs que nous avons fabriqués, nous pouvons les organiser en obstacles et leur donner des formes géométriques 3D.

Références bibliographiques

Références bibliographiques

- [1] «qu'est-ce-qu'un-robot-apprendre-a-connaître-les-capteurs-et-les-actionneurs,» diwo, 02 06 2023. [En ligne]. Available: <http://diwo.bq.com/fr/qu-est-ce-qu-un-robot-apprendre-a-connaître-les-capteurs-et-les-actionneurs/>.
- [2] «Robotique : qu'est-ce que c'est ?,» futura, 29 5 2023. [En ligne]. Available: <https://www.futura-sciences.com/tech/definitions/robotique-robotique-603/>.
- [3] N. e. i. Dib, «Mémoire : localisation et Navigation des Robots Mobiles ‘ INRM’». *Automatique & Informatique Appliquée*.
- [4] B.-A. Mordechai and . M. Francesco , "Robots and Their Applications," in *Elements of Robotics*, p. 3.
- [5] M. Fabio , «Initiation à la Robotique ,Licence Professionnelle Automatismes et Robotique,» 2022.
- [6] «How robotics are shaping today's world , types of robots :how robotics technologies are shaping today's world,» 2 6 2023. [En ligne]. Available: <https://www.intel.com/content/www/us/en/robotics/types-and-applications.html>.
- [7] «Robot industriel : définition, marques et applications,» journaldunet, 03 06 2023. [En ligne]. Available: <https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1440696-robot-industriel-definition-marques-et-applications/>.
- [8] Philippe Morel, «Robots chirurgicaux : jusqu'où va-t'on repousser les limites de la chirurgie ?,» revmed, 03 06 2023. [En ligne]. Available: <https://www.revmed.ch/revue-medicale-suisse/2012/revue-medicale-suisse-325/robots-chirurgicaux-jusqu-ou-va-t-on-repousser-les-limites-de-la-chirurgie>.
- [9] «Les robots domestiques,» etudier, 03 06 2023. [En ligne]. Available: <https://www.etudier.com/dissertations/Les-Robots-Domestiques/373103.html>.
- [10] Mattéo Toniutti et Jonathan Jonathan, Article - LES ROBOTS MILITAIRES, Collège André Chêne - Fleury-les-Aubrais, 2023.
- [11] «Robot explorateur , Les robots d'exploration et d'inspection des fonds marins,» RobotiqueTPE2013, [En ligne]. Available: <https://sites.google.com/site/robotiquetpe2013/robot-explorateur>.
- [12] «Les robots anthropomorphiques,» intelligenceartificielle.com, 29 05 2023. [En ligne]. Available: <https://intelligence-artificielle-robotique.weebly.com/les-robots-anthropomorphiques.html>.
- [13] J. Retto, SOPHIA, FIRST CITIZEN ROBOT OF THE WORLD -DOCTORAT, National University of San Marcos, November 2017.
- [14] «quels-sont-les-composants-des-robots,» robobox, 03 06 2023. [En ligne]. Available: <https://www.robobox.fr/blog/quels-sont-les-composants-des-robots/>.

Références bibliographiques

- [15] M. Moses, «2d lidar versus 3d lidar,» lidar and radar, 29 05 2023. [En ligne]. Available: <https://lidarandradar.com/2d-lidar-versus-3d-lidar/>.
- [16] "lidar sensors," sick sensor intelligence, 21 2 2023. [Online]. Available: https://www.sick.com/us/en/lidar-sensors/c/g575802?q=:Def_Type:ProductFamily&fs=10000#benefit_0.
- [17] «LiDAR,» en.study-robot, 11 06 2023. [En ligne]. Available: <https://en.study-robot.jp/sensor/lidar/>.
- [18] Berriche Achref, Les robots, Master-Laboratoire d'Informatique de Paris Nord, 2015/2016.
- [19] José Ricardo Sánchez-Ibáñez, Carlos J. Pérez-del-Pulgar and Alfonso García-Cerezo, "Path Planning for Autonomous Mobile Robots," 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/23/7898>.
- [20] B. Amina , DOCTORAT-Modélisation des environnements virtuels urbains dédiés à, 05/10/2020.
- [21] Z. R. MOUNTASSER, «Planification de chemin en temps réel basé sur RRT*,» BISKRA, 27/06/2022.
- [22] J. Adeel , "Understanding Dijkstra Algorithm - Master," *ResearchGate*, january 2013.
- [23] R. Dian et G. Lysander , «Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem,» *Journal of Physics: Conference Series*, 2020.
- [24] "Dept. of CS , RNLKWC Page 1 B.Sc(H) Computer Science 6th Semester Artificial Intelligence," 03 06 2023. [Online]. Available: <https://www.studocu.com/in/document/rashtrasant-tukadoji-maharaj-nagpur-university/computer-science-engineering/ai-programs/35078083>.
- [25] M. Zeyu, SLAM research for port AGV based on 2D LIDAR -A Dissertation presented in partial fulfillment of the Requirements for the Degree of Master in Telecommunications and Computer Engineerin, 2019.
- [26] «What is a servo motor?,» automationforum, 03 06 2023. [En ligne]. Available: <https://automationforum.co/what-is-a-servo-motor/>.
- [27] A. Allan, «Raspberry Pi Documentation,» 03 06 2023. [En ligne]. Available: <https://www.raspberrypi.com/documentation/computers/os.html#introduction>.
- [28] P. B. N. F. S. G. John Armston, «Py lidar Documentation Release 0.4.4,» 13 Jul 2022.
- [29] Sebastian, «GPIO Programming on the Raspberry Pi: Python Libraries,» medium, 30 Aug 2021. [En ligne]. Available: <https://medium.com/geekculture/gpio-programming-on-the-raspberry-pi-python-libraries-e12af7e0a812>.
- [30] «Matplotlib: Visualization with Python,» matplotlib, 05 06 2023. [En ligne]. Available: <https://matplotlib.org/>.