

GLO-4001/GLO-7021 Introduction à la robotique mobile

TP3 Version 0.1

En équipe de 1 à 2

6 décembre 2024

ATTENTION : Ce travail pratique devra être réalisé dans des [Jupyter Notebooks](#). Répondez directement dans les *Notebooks*. Pour la remise, vous devrez exporter vos *Notebooks* avec éléments demandés (figures, images, etc.) sous format PDF. Pour ce faire, utilisez le menu **File -> Save and Export Notebook As -> PDF**.

Pour les étudiants en GLO-7021, veuillez noter qu'une présentation déficiente dans le rapport (manque de clarté, orthographe et grammaire, police de caractère illisible sur figure, etc.) pourra entraîner une pénalité allant jusqu'à 10 % de la note.

Le travail est noté sur 80 pour les étudiants de GLO-4001, et sur 100 pour les étudiants de GLO-7021.

Introduction

Dans ce projet, vous aurez l'occasion de travailler avec des données issues de capteurs robotiques réels, en particulier des données de LiDAR, pour accomplir plusieurs tâches inspirées par le DARPA Subterranean Challenge. Ce défi met l'accent sur l'exploration et la navigation autonomes dans des environnements complexes et inconnus, ce qui est directement applicable à de nombreuses missions robotiques réelles, comme la recherche et le sauvetage ou l'exploration souterraine. Une partie du défi demandait de localiser des objets d'intérêts et de reporter leur position dans une carte construite par le robot. Voici une vidéo résumant le défi ainsi que la performance du Norlab de l'Université Laval : [CRV2020 - From subterranean to subarctic autonomous exploration](#).

Dans le cadre du travail pratique, nous allons utiliser la plateforme robotique Wave Rover de la compagnie Waveshare utilisée dans les laboratoires du cours. Toutes les plateformes disposent d'un ordinateur de bord Jetson Orin, une caméra OAK-D Lite, un IMU intégré et un capteur LiDAR 2D.

Vous commencerez par cartographier une mini-ville composée de blocs, à partir des données LiDAR fournies. Pour aligner les scans et créer une carte cohérente, vous utiliserez l'algorithme Iterative Closest Point (ICP). Une fois la carte générée, vous localiserez des objets d'intérêt représentés ici par des marqueurs fiduciaires [ArUco](#).

Enfin, en utilisant une représentation sous forme de grille d'occupation de la carte, vous implémenterez un algorithme de planification de chemin, le Rapidly-exploring Random Tree (RRT). Cet algorithme vous permettra de trouver un chemin tout en évitant les obstacles présents dans l'environnement.

Prérequis

Conseils

Avant de commencer le TP3, lisez bien **tout le document**. Les questions ont plusieurs éléments en commun, il est préférable de tous les avoir en tête avant de coder. Ce travail pratique devra être réalisé dans des [Jupyter Notebooks](#). Répondez directement dans les *Notebooks*. Pour la remise, vous devrez exporter vos *Notebooks* avec éléments demandés (figures, images, etc.) sous format PDF. Pour ce faire, utilisez le menu **File -> Save and Export Notebook As -> PDF**.

Installation des dépendances Python

Nous fournissons un fichier `requirements.txt` qui contient toutes les bibliothèques nécessaires, comme NumPy et OpenCV (pour la détection des marqueurs fiduciaires). Exécutez la commande suivante pour installer les dépendances.

```
pip install -r requirements.txt
```

Lancer le *Notebook*

Pour lancer le [Jupyter Notebooks](#), exécutez la commande suivante.

```
jupyter notebook
```

Exporter un *Notebook* en PDF

Pour la remise, vous devrez exporter vos *Notebooks* avec éléments demandés (figures, images, etc.) sous format PDF. Pour ce faire, utilisez le menu **File -> Save and Export Notebook As -> PDF**.

Description de la collecte de données

Pour la collecte de données, nous avons utilisé les *Notebooks* disponibles dans les laboratoires du cours [norlab-ulaval/glo4001/collect](#). Les données sont sous format [pickle](#).

1 Cartographie à partir de scans LiDARs

(40 points pour GLO-4001, 50 points pour GLO-7021)

Dans cette section, vous allez résoudre un problème de cartographie en utilisant des scans LiDAR fournis par un robot mobile. L'objectif est de construire une carte de l'environnement en combinant plusieurs scans LiDAR et d'y localiser des marqueurs ArUco.

ATTENTION : Votre position initiale est l'origine de votre référentiel. Nous considérons le centre du LiDAR comme le centre du référentiel du robot. L'axe Y positif correspond à l'avant du robot et l'axe X positif à la droite.

Pour cette section, on vous fournit les données de scans LiDARs (`lidar/lidar_00001.pkl`) et des images de la caméra Oak Lite (`oak/oak_00001.pkl`).

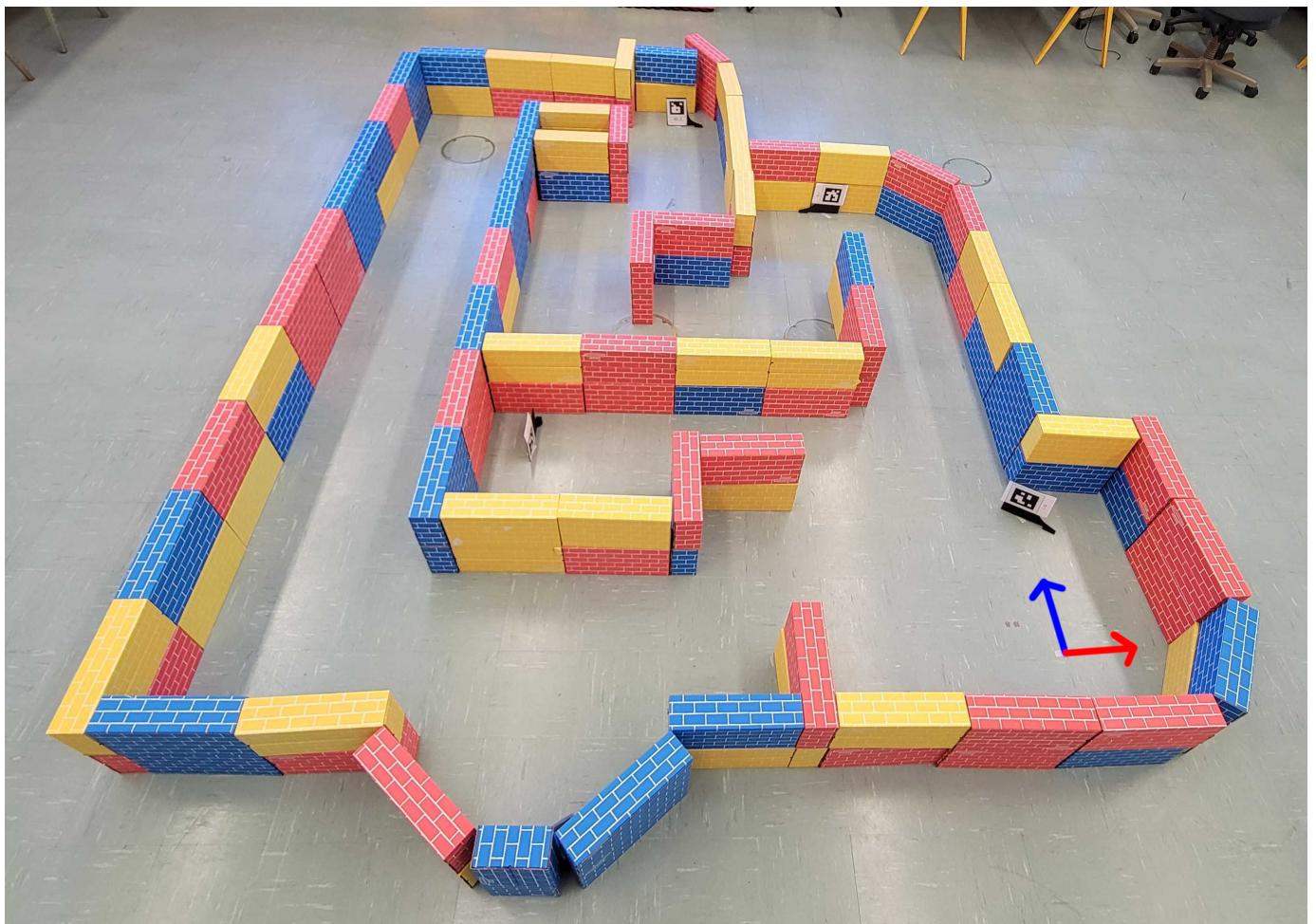


FIGURE 1 – Photo de la ville utilisée pour la cartographie. Le référentiel de la file est dessiné dans le coin inférieur droit. La flèche rouge indique l'axe X positif, et la flèche bleue l'axe Y positif. Ainsi, le LiDAR de la plateforme robotique est placé à l'intersection des flèches.

1.1 Détection de marqueurs fiduciaires Aruco (5 points)

À cette étape, vous allez détecter des marqueurs fiduciaires ArUco dans les images capturées par la caméra du robot et transformer leur position dans le référentiel de la caméra en coordonnées 2D dans le référentiel du LiDAR.

1.2 Recalage de deux nuages de points (15 points)

Dans cette étape, vous allez recaler ensemble deux nuages de points à l'aide de l'algorithme ICP. Cet algorithme permettra d'estimer une transformation qui aligne au mieux les deux nuages de points. On ne vous demande pas d'implémenter ICP, mais plutôt d'utiliser une implémentation fournie avec le code de base qui se trouve dans le dossier `r2_lidar_icp`. Le code provient de [willGuimont/r2_lidar_icp](#) et a été copié avec la permission de l'auteur.

Vous devriez remarquer que de simplement appliquer ICP ne fonctionne pas. Comment pouvez-vous résoudre ce problème ?

1.3 Cartographie (10 points pour GLO-4001, 20 points pour GLO-7021)

Dans cette partie, vous allez construire une carte de la mini-ville en utilisant plusieurs nuages de points capturés par le robot. L'objectif est de recaler l'ensemble de ces nuages pour obtenir une vue cohérente de l'environnement.

1.4 Localisation des marqueurs dans la carte globale (10 points)

Maintenant que vous avez à la fois la position des marqueurs dans le référentiel du robot et la pose du robot dans la carte globale, vous pouvez trouver la position des marqueurs dans la carte globale.

2 Planification de chemins (40 points pour GLO-4001, 50 points pour GLO-7021)

Dans cette section, vous allez mettre en œuvre la planification de chemins dans une grille d'occupation. L'objectif de la question est d'implémenter RRT et RRT-Connect pour faire de l'exploration et de la planification de chemin.

Pour cette section, on vous fournit des grilles d'occupation `world.png` et `world_q1.png`. Ces cartes sont des images dans lesquelles les pixels noirs représentent des obstacles et les pixels blancs sont des espaces libres.

2.1 Implémentation de RRT (20 points)

L'algorithme RRT génère un arbre de recherche aléatoire à partir du point de départ, en étendant progressivement les branches vers des zones non explorées. Dans cette section, vous devrez implémenter l'algorithme et le tester sur `world.png`.

2.2 Implémentation de RRT-Connect (15 points)

L'algorithme RRT-Connect construit sur RRT en créant deux arbres qui croissent l'un vers l'autre dans le but de faire de la planification de chemin. Une fois que les arbres se touchent, il est possible de retrouver un chemin reliant le point de départ au point d'arrivée. Dans cette section, vous devrez implémenter cet algorithme et le tester sur `world.png`.

2.3 Analyse (5 points pour GLO-4001, 15 points pour GLO-7021)

Cette section inclut des questions d'analyse sur les algorithmes RRT et RRT-Connect. Vous devrez analyser les différents compromis entre les paramètres et générer votre propre grille d'occupation dans laquelle tester vos algorithmes.