PFE n°9 : SlamIT

Lina MEJBAR - IMA 5

MADE WITH
beautiful.ai

SOMMAIRE

1. Contexte
2. Robot et équipements
3. Cartographie
4. Web

SLAMIT

MADE WITH beautiful.ai

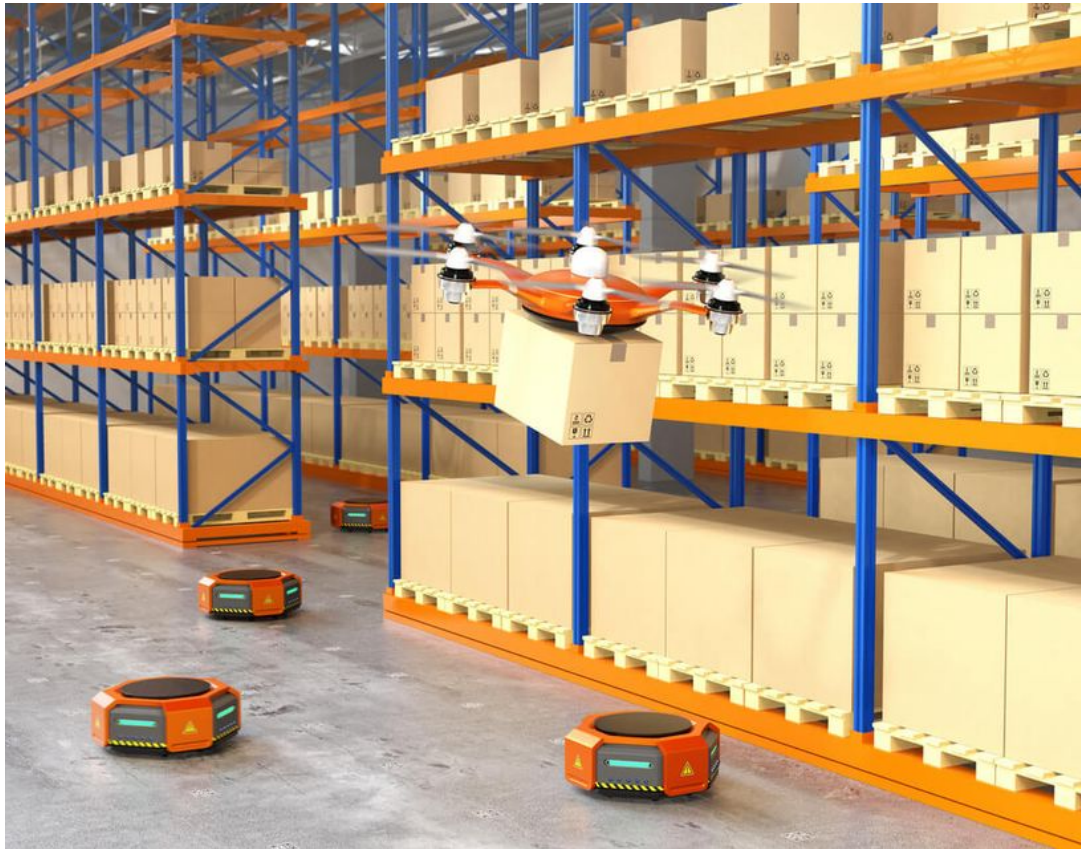
1 - Présentation du projet

Présentation du projet

- Bonduelle
- Contexte
- Première phase de recherche

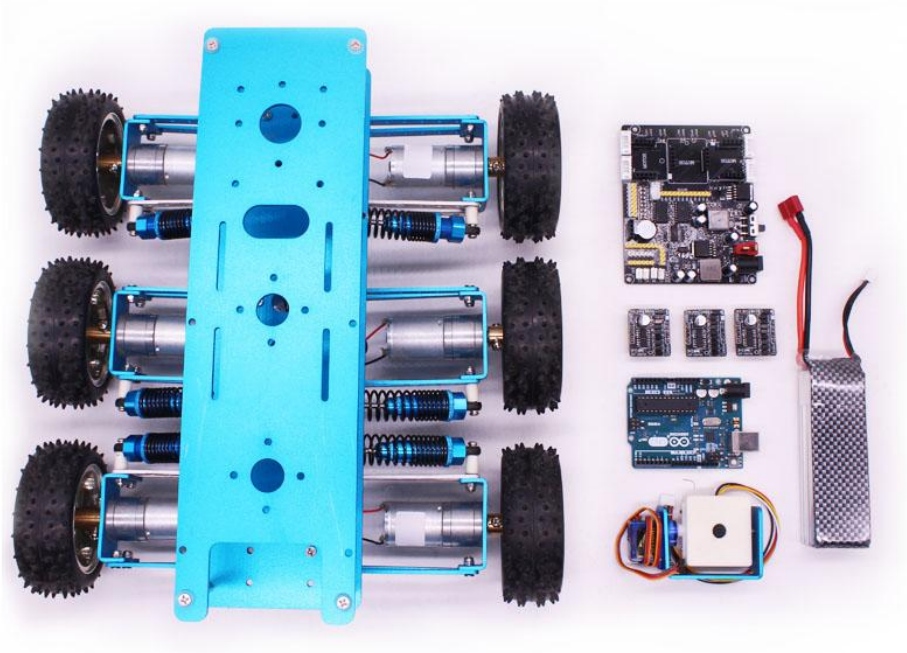
Cahier des charges

- **Gérer le robot à distance**
  - Son état
  - Son déplacement
- **Cartographier en continu la pièce**
  - Gestion des obstacles mobiles
- **Déterminer un chemin optimal**



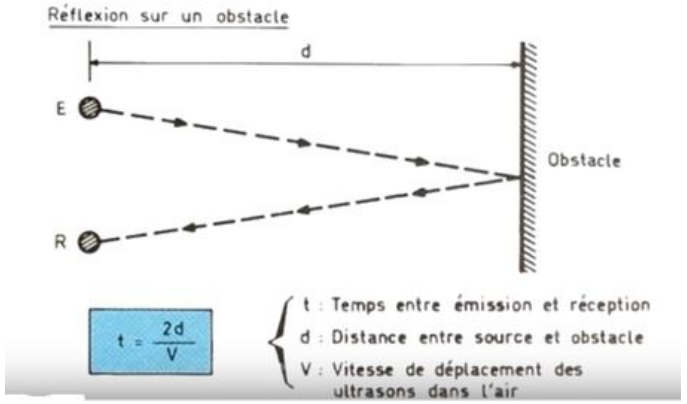
2 - Robot et équipements

Equipements du projet



- **Marque** : Yahboom
- 6 moteurs, roues
- Cartes
  - Extensive
  - Arduino
  - Raspberry Pi 4
- Lidar
- Coral
- Encodeurs

Lidar



Abréviation :

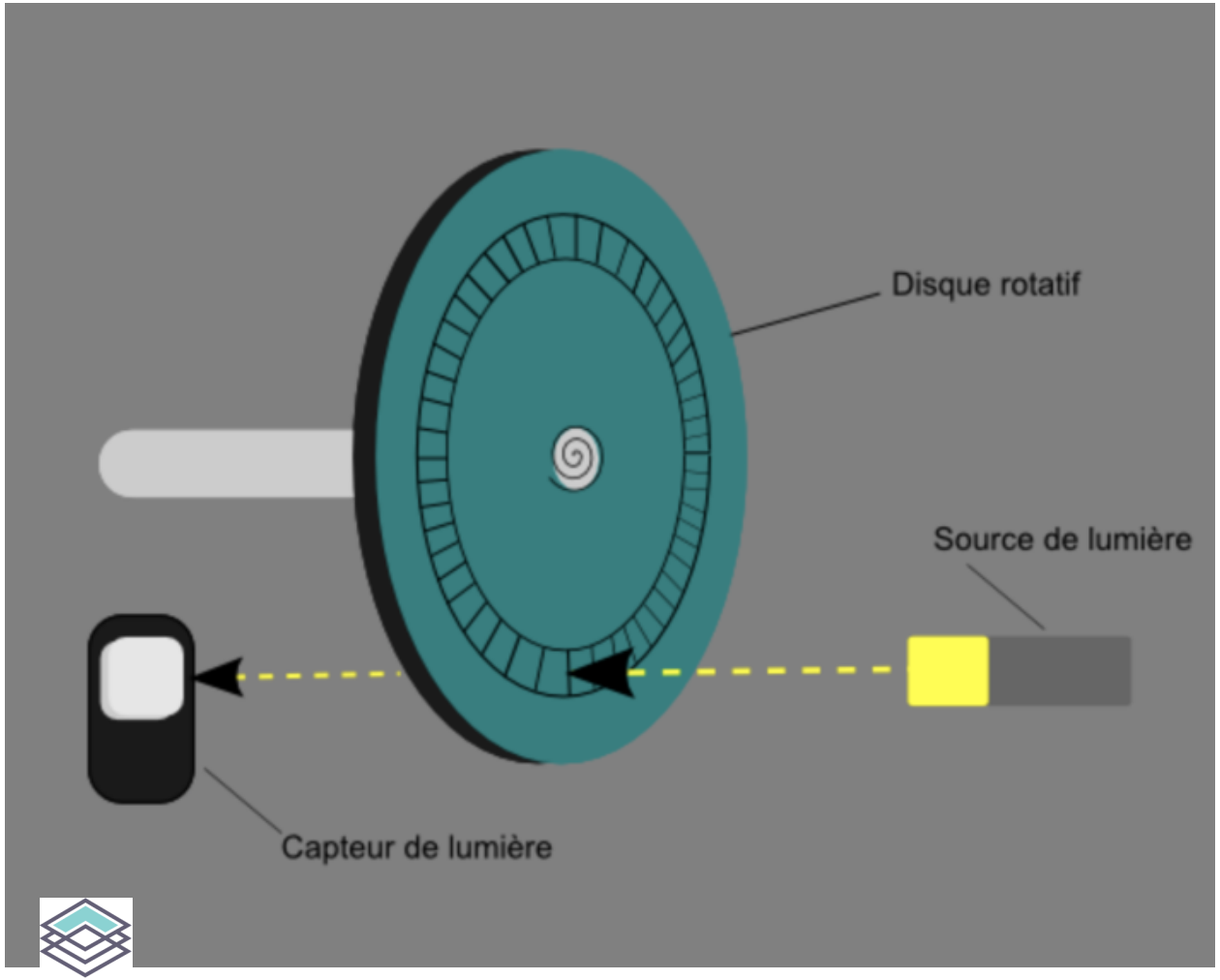
LIght Detection And Ranging

Principe :

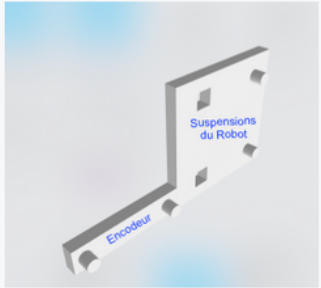
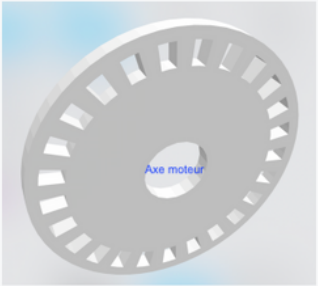
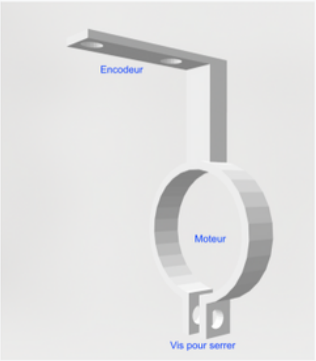
- Calcul du temps entre l'envoi du faisceau laser et sa reception => équivalent à un ultrason
- Rotation 360°

Encodeurs

Principe :



Modélisation :

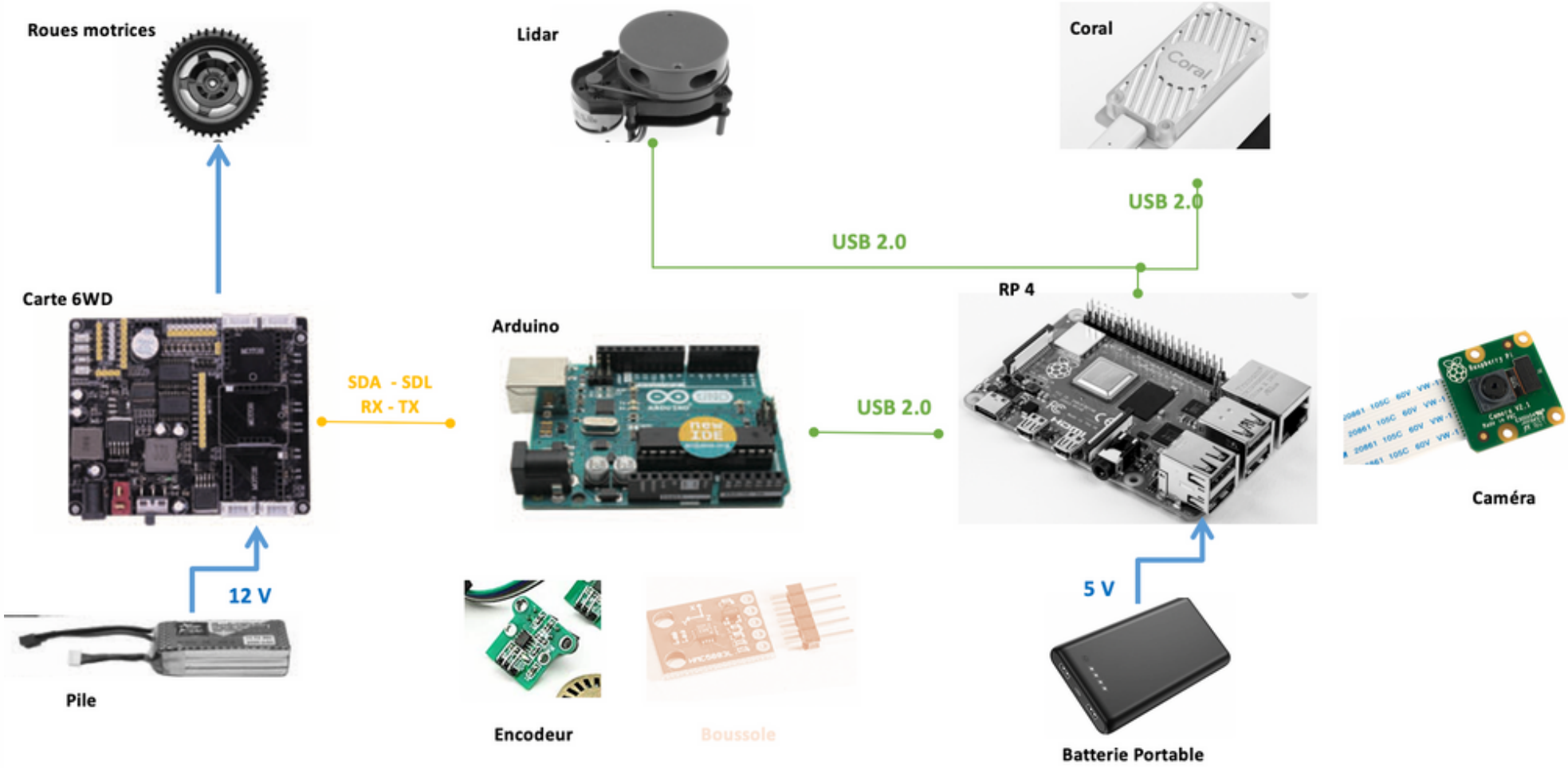
Modélisation 1	Modélisation 2	Modélisation 3
Rallonge axe moteur	**Support encodeur**  **Disque de l'encodeur** 	**Support encodeur v2**  **Disque de l'encodeur v2**

Coral

Principe :

- Accélérateur USB
- Puce Edge TPU (Tensor Processing Unit)
- Optimisé pour les applications de ML
- 4 billions d'opérations /s (tops)
- Consommation : 0,5W pour chaque sec (2 tops/W).

Résumé des équipements du projet



3 - Cartographie

Cartographie - OpenCV

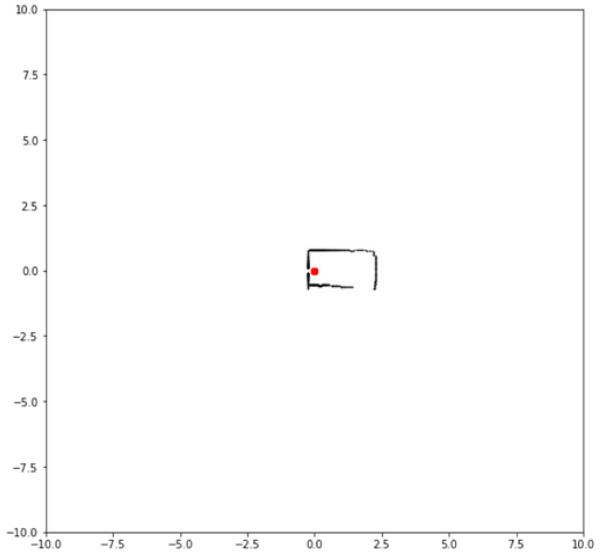
- Open Computer Vision

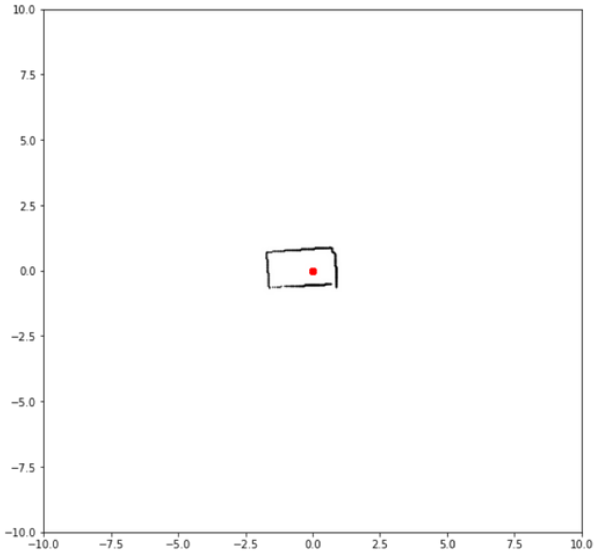
- Bibliothèque libre de traitement d'images en temps réel

Exemple d'utilisations :

- Lecture, écriture affichage d'une image
- Calcul de l'histogramme, niveaux de gris,
- Lissage, filtrage
- Seuillage d'image

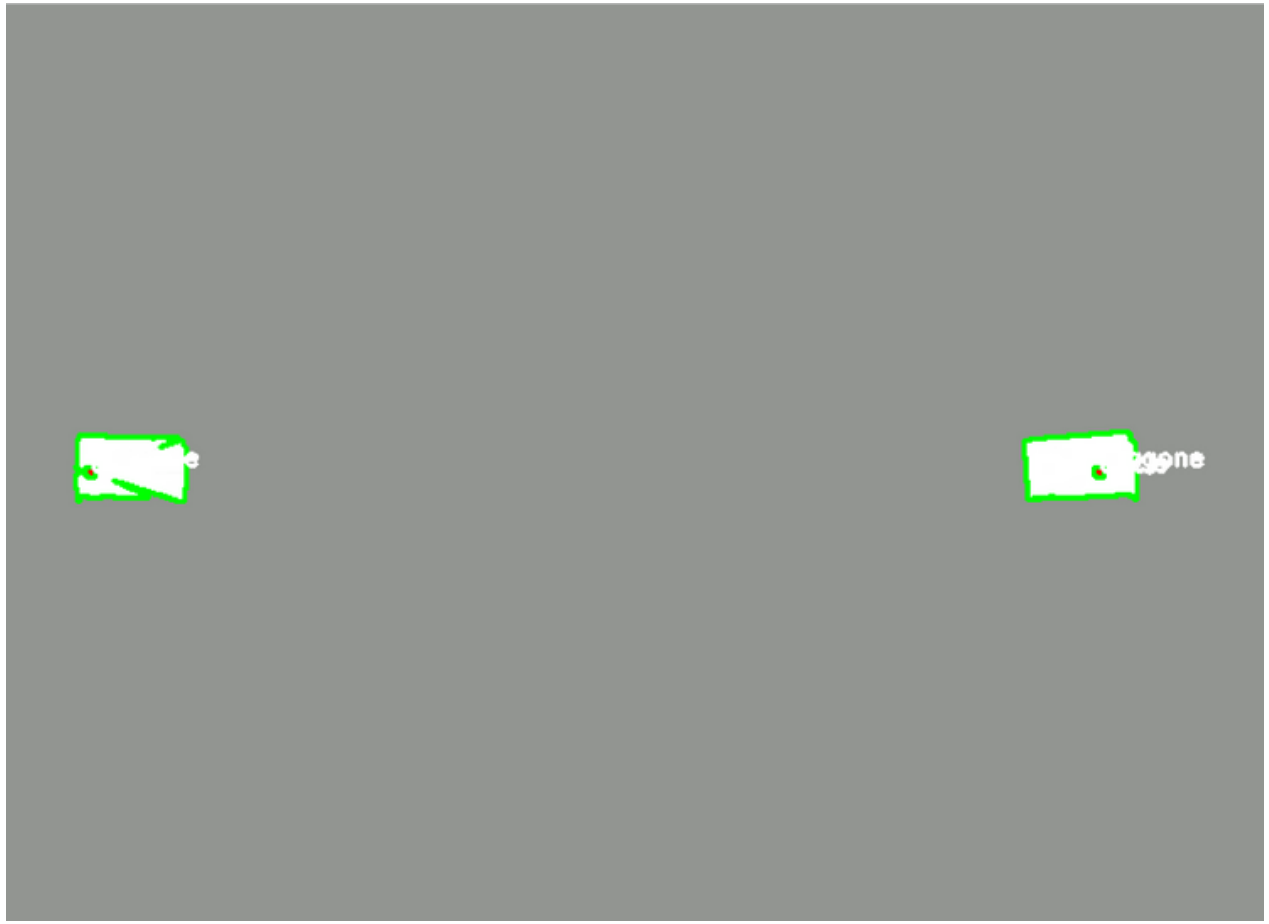
Cartographie - OpenCV





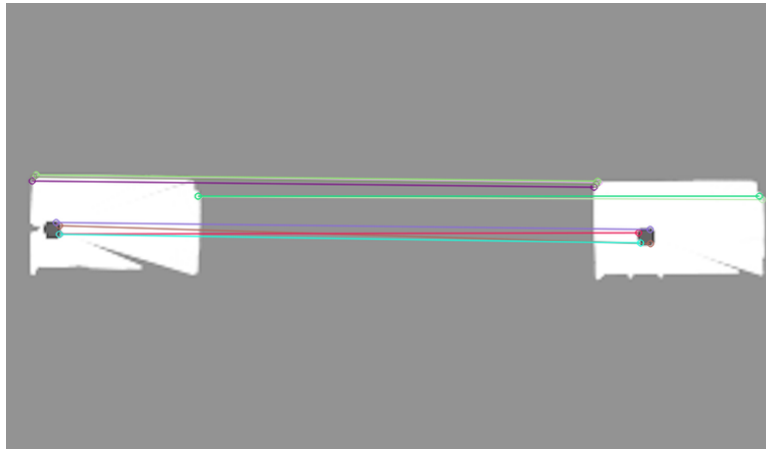
- Pas de détection de forme
- Pas de détection de ligne
- Pas de détection de similarité

Cartographie - OpenCV

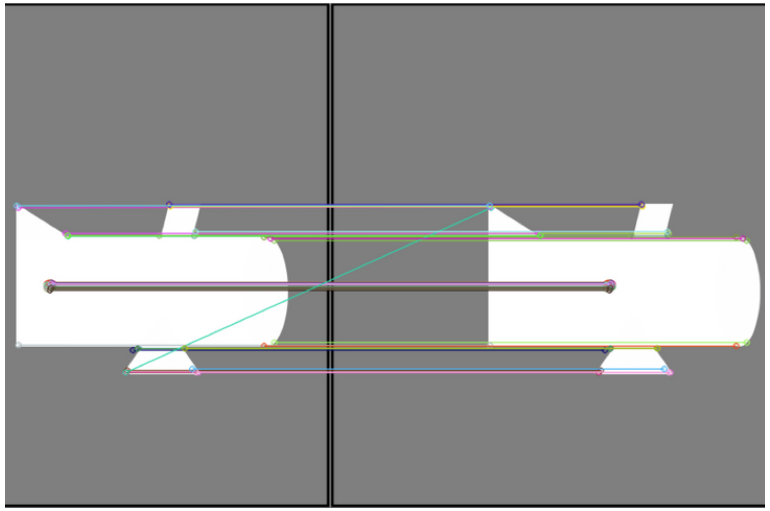


- Détection de formes

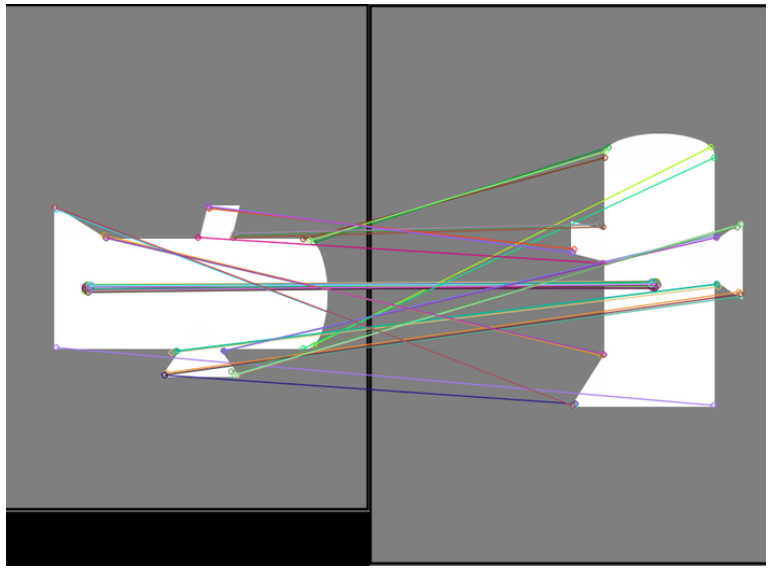
Détection des similitudes - OpenCV



Deux cartographies



Forme créée

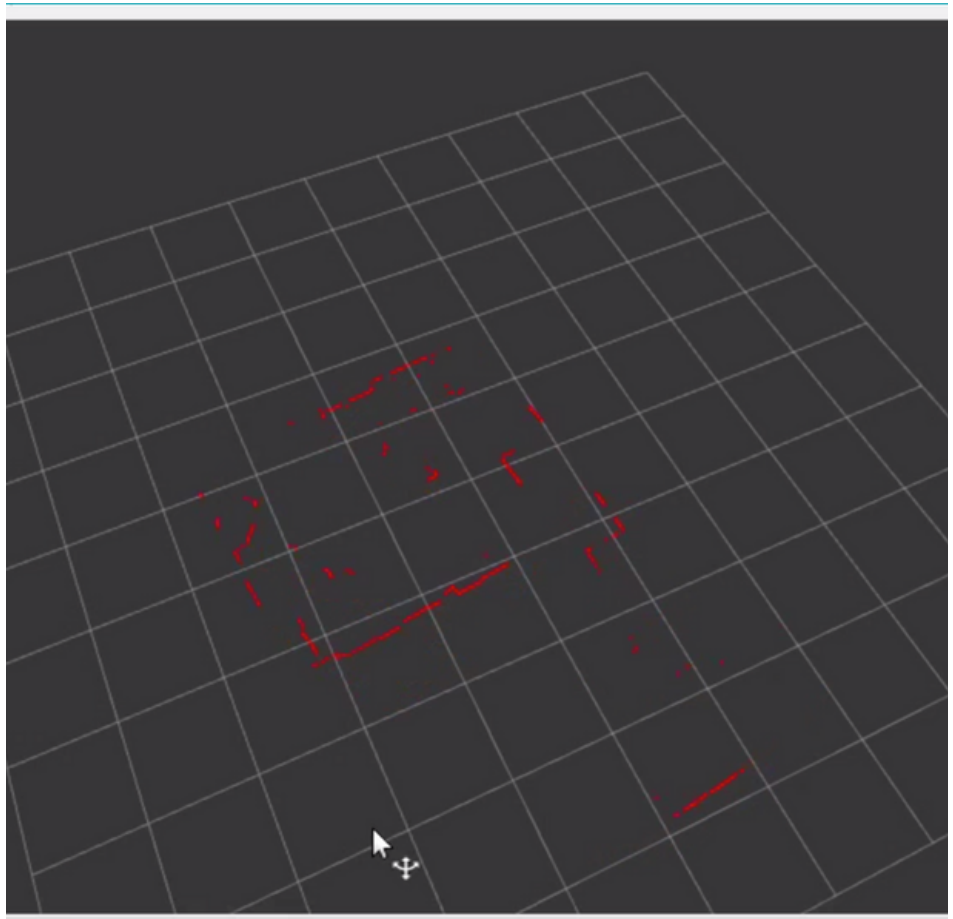


Forme créée avec une rotation

Cartographie - ROS

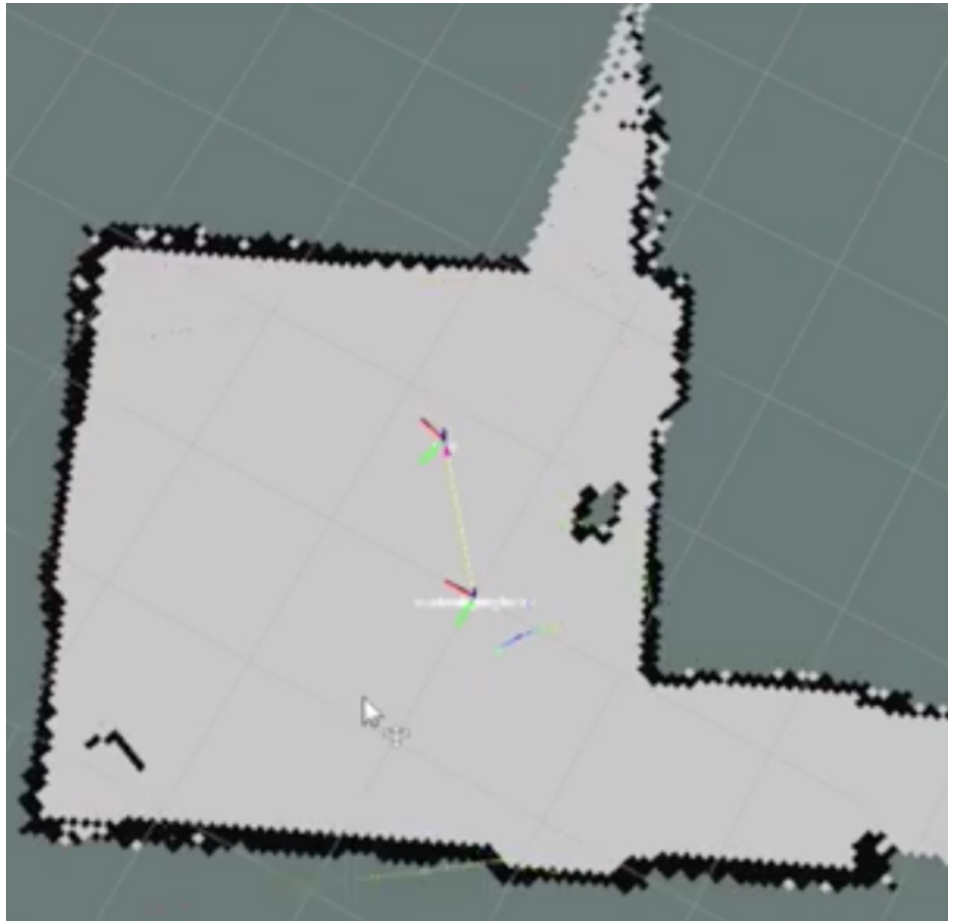
- Robot Operating System
- Ensemble d'outils informatiques open source pour la robotique

ROS - YDlidar



- Placement des points

ROS - Hector Slam



- Construction de la carte en temps réel
- Correction de la carte en autonomie

ROS - Hector Slam

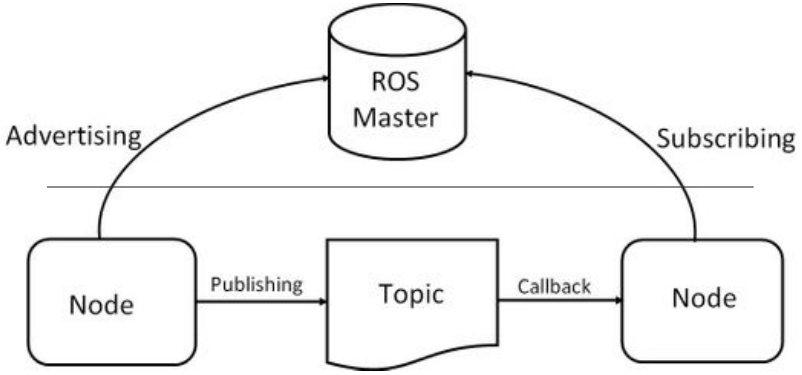
Video

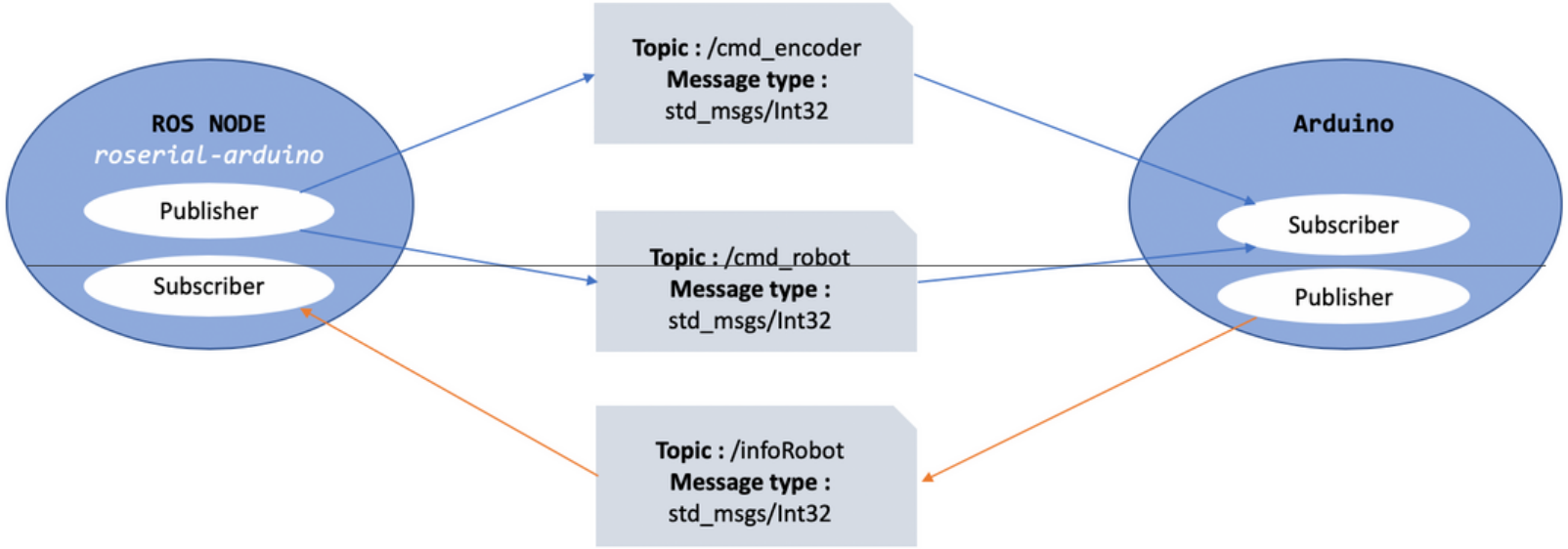
ROS - Commande du robot

```
pi@raspberrypi      :~$ rosmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

- Message Twist
- ROS Serial

Fonctionnement de ROS





4 - Web

Site du projet - *Fonctionnalités*

Robot :

- Déplacer le robot

Cartographie :

- En temps réel

Caméra :

- Reconnaissance d'objets en temps réel

Récapitulatif - *Travail réalisé*

Récupération des données du Lidar

Utilisation de la librairie : YLIDAR

Déplacement du robot à distance

Utilisation de Websocket

Réalisation du site internet

HTML - CSS - PHP - ROS2DJS

Caméra et IA

- Utilisation de Coral pour faire de la détection d'objet
- Framwork Flask pour publier le flux vidéo

Retour expérience du projet

- Réaction face aux aléas
- Amélioration du projet
- Compétences

Merci pour votre attention

Remarques / Questions

MADE WITH beautiful.ai