



UNIVERSITÉ DE MONS

TP4

---

## Défis en Intelligence Artificielle

---

*Auteur :*  
Maxime De Wolf  
Dimitri Waelkens

14 janvier 2019

## 1 Configuration de *Cartographer*

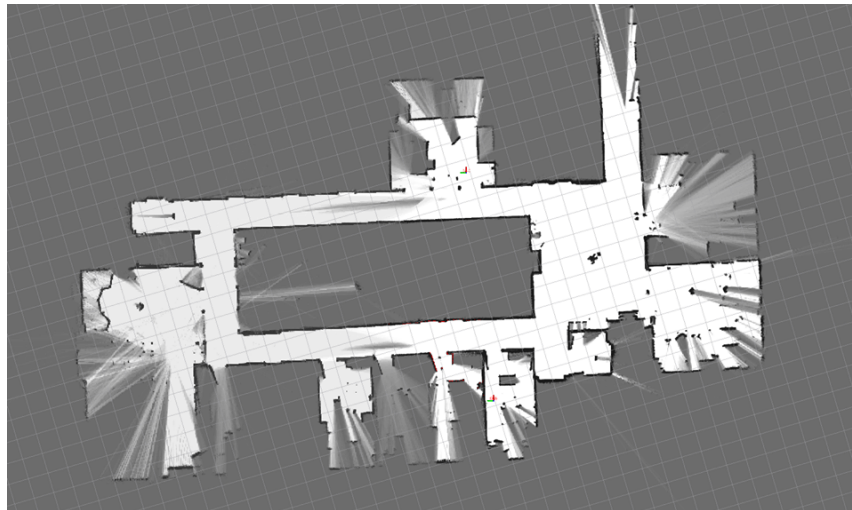


FIGURE 1 – Carte obtenue grâce à *Cartographer*

## 2 Localisation des objets

Cette section présente les résultats obtenus pour la détection d'objets que le robot à filmé. Nous expliquons donc dans un premier temps la logique que nous avons suivies pour obtenir ces résultats. Nous justifions ensuite la manière dont ils pourraient être améliorés.

### 2.1 Technique utilisée

Pour résoudre ce problème, nous utilisons une technique très simple. Effectivement, à chaque fois que le robot détecte un objet, nous repérons la position du robot ainsi que la distance à laquelle l'objet a été filmé. Cependant, nous avons également besoin de l'orientation du robot pour calculer la position de l'objet.

Afin d'avoir l'orientation du robot, nous repérons également l'emplacement précédent duquel le robot voyait également cet objet. Nous pouvons donc calculer former par la droite passant par ces deux points dans le repère.

Ensuite, grâce à la distance de l'objet filmé, nous pouvons déterminer la position de celui-ci.

Le robot travaille sur un grand nombre d'image similaire. Nous calculons donc plusieurs fois la position du même objet.

Pour pallier à ce problème nous utilisons un algorithme de *clustering* simple afin de regrouper les position qui sont susceptible de représenter un même objet.

Cette algorithme fonctionne de la manière suivante :

1. Pour chaque position, on regarde si elle est proche du centre d'un *cluster* existant ;
2. Si c'est le cas, on l'ajoute à ce *cluster* ;
3. Sinon, on crée un nouveau *cluster* qui contient cette position ;

Finalement pour retrouver la position de chaque objet, on calcule le centre de chaque *cluster*. Cette technique nous donne les résultats exposés par la Table 1.

Objet	Position 1	Position 2
<i>teddy bear</i>	(2.399926850060335, -0.008283207609849293)	(12.604331235659147, 14.238862468344005)
<i>motorbike</i>	(8.366541118657906, 14.943175542227172)	/
<i>backpack</i>	(4.832979494372698, 2.7004257263585636)	/
<i>suitcase</i>	(5.446312729568893, 16.221012094608028)	/
<i>pottedplant</i>	(12.343678479509247, 14.322766597693517)	(0.9148871784197752, 23.25876369786711)
<i>stop sign</i>	(7.87594829417553, 15.93296178462322)	/
<i>refrigerator</i>	(6.239195085908132, 13.789541846887905)	/

TABLE 1 – Positions obtenues pour chaque objet détecté

## 2.2 Discussion sur les résultats

Grâce à la Table 1, nous pouvons voir que nos résultats sont assez mauvais par rapport aux résultats attendus car nous ne détectons pas le bon nombre d'objets. Nous dressons donc une liste non-exhaustive de tous les facteurs qui pourraient être responsables.

Pour retrouver la position du robot précédant la détection de l'objet, nous prenons la position du robot durant la détection précédente d'un objet. Cela implique que nous prenons éventuellement la position du robot pendant la détection d'un autre objet. Cela peut donc causer la création de *clusters* artificiel.

Une solution pour régler ce problème est de tester si l'objet précédemment détecté est le même que celui en cours de détection. On pourrait notamment faire ceci en se basant sur la technique de *clustering* que nous avons utilisé.

Une autre solution serait de ne plus prendre la position du robot pendant la détection de l'objet précédent mais de sélectionner la position du robot juste avant de détecter l'objet. Cette solution est selon nous la plus logique à appliquer.

Notre technique de *clustering* est très (trop?) simple. Dès lors, on peut se demander si d'autres techniques donneraient de meilleurs résultats. Effectivement, chaque position que nous trouvons découle de la formation d'un *cluster*. Un problème à ce niveau aurait donc des conséquences directes sur nos résultats au niveau du nombre d'objets détectés.

Le robot nous fournit la distance à laquelle se trouve ce qu'il voit. Cependant, il arrive qu'il détecte plusieurs objets sur la même image. Dans ce cas nous utilisons la même distance sur tous ces objets ce qui implique que tous les objets présents sur une même image auront d'office la même position. Or, cela est loin d'être vrai. Ce problème influe donc sur la position des objets détectés, ce qui a une incidence sur les *clusters* formés et donc sur les résultats finaux.