Projet | Cahier des charges

Gaspard Ducamp et Yoann Taillé – UPMC – 2016

Approche topologique pour la planification de mouvements
(d'un robot de type T-800)

Introduction

"Où est Sarah Connor?" – et comment se rendre à elle?

Venu tout droit d'un futur apocalyptique où les machines cherchent à amoindrir la population humaine, le robot T-800 essaie de sauver Skynet, la société qui l'a produit. Pour cela, il doit se rendre à Los Angeles en 1984, pour trouver et "éliminer" Sarah Connor, dont le fils ne sera nul autre que John Connor, futur chef de la rébellion humaine.

Ayant déjà effectué une recherche dans les Yellow Pages, il a réussi à déterminer où travaille Sarah, et en s'adressant au service informatique de l'entreprise et en promettant à ses membres une FAMICOM, il parvient à en obtenir les plans. Les moyens technologiques étant limités à l'époque et ne sachant pas à quoi elle ressemble, il est forcé à interroger les autres employés, d'où la fameuse question.

Dans les années 80, les robots se déplaçaient sans prendre en compte la nature de leur environnement ni des obstacles qu'ils peuvent rencontrer, contrairement au Terminator...

Nous cherchons dans ce projet à planifier les mouvements de notre robot (assimilé à un cercle) à travers un immeuble comportant des étages de plusieurs pièces. Ceux-ci étant occupés, du mobilier est éparpillé à travers les pièces, représentant des obstacles sur le trajet du robot. En supposant que la position de la cible est connue, notre objectif est donc de déterminer un chemin (en 2 dimensions) efficace à travers les différentes pièces pour que l'androïde l'atteigne sans heurter les obstacles (assimilés à des polygones).

Spécifications techniques

Outils de programmation

- Langage principal : Python
- Outils secondaires : XML, Tkinter, Pyplot, Sphinx, Pyglet

Interface

L'utilisation d'une interface graphique sera suggérée à l'utilisateur afin de lui permettre de visualiser son espace de travail ainsi que les étapes permettant le calcul du trajet du robot dans son environnement.

Mise en oeuvre

Données d'entrée

L'utilisateur devra fournir au programme un fichier au format .xml contenant les diverses informations qui peuvent caractériser l'environnement de travail. Un schéma xml sera fourni afin de respecter les règles de syntaxe mises en place. La description devra se faire de la manière suivante, pour chaque pièce devront être fournis :

- Un identifiant unique.
- La position des différents murs (de manière relative à elle-même, le choix de l'origine est à la convenance de l'utilisateur).
- Une description des différents obstacles qu'elle comporte, c'est-à-dire la position des segments les composant.
- L'emplacement des différentes sorties de celle-ci ainsi que les identifiants des pièces adjacentes rendues accessibles.

Des fichiers caractérisant les contenus des pièces seront alors créés, il sera possible de modifier "à la volée" la position des obstacles (il faudra cependant vérifier respecter l'unicité des identifiants). L'utilisateur devra ensuite définir les positions de départ et d'arrivée du robot, ainsi que son rayon (qui correspond à sa largeur).

Première étape

Dans un premier temps notre programme se chargera d'identifier, et cela grâce à un algorithme de type "plus court chemin", dans notre cas celui de Dijkstra, les pièces devant être parcourues de manière à relier la position initiale et la position but de la manière la plus efficace sans prendre en compte les obstacles. Ce faisant une économie de calculs sera faite.

Seconde étape

Le programme tâchera ensuite de calculer de manière séquentielle les différents chemins (porte à porte) dans chacune des pièces, cette fois en évitant les obstacles. Pour se faire nous utiliserons un algorithme non conventionnel nous permettant de manipuler l'espace en s'inspirant d'une mathématique consubstantielle à celle des difféomorphismes. Il consiste à réduire géométriquement, grâce à une fonction difféomorphe, les obstacles présents dans les pièces, afin de tout d'abord simplifier le déplacement du robot le long d'une ligne droite. Une fois les déformations effectuées, il s'agira d'appliquer à la ligne leurs inverses, renvoyant ainsi un chemin respectant la topologie de la pièce.

Si la présence d'obstacles infranchissables nous empêche de trouver un chemin réalisable, il serait envisageable d'implémenter un retour arrière nous renvoyant dans une configuration précédente.

Par ailleurs, l'utilisation de déformations, contrairement aux méthodes classiques d'arbres de planification stochastiques, permet une compréhension topologique de l'espace, ce qui pourrait également simplifier la résolution d'autres types de problèmes, de type patrouilles ou "couverture de zones entières à la Roomba".

Données en sortie

A la fin de la deuxième étape le programme affichera la trajectoire calculée pour le robot, celle-ci sera enregistrée dans un fichier sous la forme d'une suite de coordonnées.