Developer Guide: Representation and relative positioning from visual information

Realized by:
Asma BRAZI

Proposed by:
Cédric HERPSON

Laboratory of Computer Sciences, Paris 6
Sorbonne University - Faculty of Science and Engineering

June-July 2019

Contents

Chapter 1

Présentation du code

1.1 La partie robot

1.1.1 Python

Le premier language utilisé est le python, qui est executé sur le Raspberry PI. Nous avons utilisé ce language car il était déjà utilisé par le groupe d'avant, mais il peut être modifié par du C pour gagner en performance.

La première chose que fait le programme en Python est le traitement d'image, la librairie OpenCV. Cette partie nous permet de faire de la reconnaissance d'image, mais nous avons aussi calculer les lignes de fuites d'images, qui nous sert à replacer le robot.

Nous utilisons aussi le python pour pouvoir gérer facilement des fichiers. Nous écrivons régulièrement des données dans des fichiers, qui seront ensuite envoyés vers le PC pour mettre à jour le modèle 3D.

Pour finir, nous utilisons aussi le python pour modifier les fichiers Aseba, qui sera ensuite compilé pour être utilisé pour le Thymio.

1.1.2 Aseba

Comme précisé ci-dessus, le Thymio utilise du code Aseba pour se déplacer. Nous ne programmons cependant pas en Aseba directement, nous le faisons avec Python. Il est cependant utile de connaitre les bases en Aseba pour programmer dans les meilleurs conditions.

1.2 La partie ordinateur

Pour la partie du code dédié à l'ordinateur, nous utilisont du Java. Java nous permet d'abord à communiquer avec le Raspberry PI, notemment en récupérant les données qu'il écrit pendant son exploration.

Ensuite, Java nous permet aussi d'avoir un retour vidéo de l'exploration du robot. A l'aide de la librairie JMonkey Engine nous pouvons afficher une modèle 3D que le robot construit pendant son exploration.

Pour finir, nous utilisons aussi Java pour écrire les fichiers python sur le Raspberry PI, ce qui nous oblige pas à copier manuellement les fichiers à chaque fois que nous les modifions.

Chapter 2

Documentation interne

La documentation interne du code que nous fournissons ne concerne que les fonctions que nous avons ajouté ou modifié. Pour en savoir d'avantage sur les autres fonctions du code, il est préférable de se référé à la documentation interne des étudiants de l'an dernier.

2.1 Python

Fichier imageRecognition

imageRecognition(string imageURL, string imageName)
Cette fonction permet de reconnaître un des repères de la base de données dans l'image donnée en paramètre. Elle permet aussi de calculer la distance à ce repère.
Renvoie: double distance

Fichier imageUtil

normalise(np.array image)
Calcule la matrice passée en paramètre, normalisée entre 0 et 255.
Renvoie: np.array normImage

resize(np.array image, int width)
Redimensionne la matrice donnée en paramètre en une matrice dont la taille est proportionnelle à la largeur de l'image divisé par width.
Renvoie: np.array resizedImage

fichier imageVanishingLines

Ce script calcul l'angle moyen des lignes de fuites horizontales pour calculer l'angle par rapport au mur. Renvoie: l'angle par rapport au mur.

fichier initWorkspace

Ce script permet de créer un dossier DB dans lequel sera déplacé toutes les images.

Fichier utils

HoughLines(string im, int maxLG)
En utilisant les lignes de fuite verticales, cette fonction calcule la hauteur du mur sur l'image im. La paramètre maxLG est la valeur maximale que nous autorisons pour cette hauteur. Renvoie: la hauteur du mur.

takePicture(int r1, int r2)
Permet de prendre une photo, avec la résolution r1 et r2. Renvoie: l'url de l'image

getThreshold(type)
Renvoie un score limite selon de type de photo pour considérer qu'une image est reconnue ou non dans l'image. Renvoie: le score correspondant.

recognition(string image, string ref, string multi, string nbRefs)
Cette fonction permet de reconnaître les objets de la base de donnée dans l'image dont l'url est passée en paramètre. Si multi est à 'm', alors on cherchera parmis les nbRefs objets de la base de donnée. Sinon, on cherchera seulement l'objet ref de la base de donnée. Renvoie: Une liste contenant la distance à chaque objet et leurs coordonnées si les images sont trouvés, sinon des valeurs négatives.

2.2 Java

2.2.1 Package Representation

Classe Scene

simpleUpdate()
Met à jour tous les elements de la scène.

initKeys()
Attribut des fonctions que le robot devra effectuer à certaines touches du clavier.

onAction()
Appelle des fonctions lorsque les touches attribuées sont appuyées.

updateWall(wall w)
Met à jour le wall passé en paramètre en modifiant sa taille et son angle.

2.2.2 Package exploratior

Classe Exploration

start()
La méthode appelée au début de l'exploration. Elle demande à l'utilisateur quel est l'objet à recherché, et va lancer la méthode *explore* sur les murs de la pièce.

explore(int facingWall, int target)
C'est dans cette méthode que l'algorithme d'exploration a été implémentée. Le robot va alors faire son exploration sur le mur d'ID facingWall jusqu'à trouver l'objet dont l'indice est égal au paramètre target, ou alors arriver jusqu'au bout du mur. L'ID du mur que nous explorons permet aussi de mettre à jour le bon mur dans le représentation 3D de l'environnement.
Renvoie: 1 si l'objet est trouvé. 0 si on a atteint le coin du mur. -1 si on a détecté un objet ne faisant pas parti de la base de donnée.

readDBObjects()
Cette méthode permet de récupérer tous les les objets de la base de donnée. La base de donnée est le fichier *dataBase*.

getINdexOfPreviousWall()
Permet d'obtenir l'indice du mur précédemment exploré.
Renvoie: l'indice du mur précédent.

2.2.3 Package envStructures

Classe InternalRepresentation

createa4WallsRoom()
Initialise la représentation 3D de l'environnement en créant 4 points qui représenteront les 4 coins de la pièce, et les murs associés.

2.2.4 Package Actionners

Classe PiThymioRobot

captureData(int type,boolean init,String multi)
Cette méthode ordonne au robot de prendre une photo et d'en extraire les données importantes. L'argument multi permet de procéder à une reconnaissance d 'image uniquement pour un coin (avec la valeur u) ou de tester toutes les images de la base de donnée (avec la valeur m). Les arguments type et init sont utilisés pour la commande envoyée au raspberry PI. Renvoie: Une liste contenant la liste d'information des objets trouvés (taille, hauteur, coordonnée).

captureNewAngle()
Cette méthode ordonne au robot de prendre une photo et de calculer l'angle par rapport au mur. Cette fonction n'est cependant plus utilisée. Renvoie: l'angle de rotation pour être face au mur.