

RAPPORT DU PROJET DU SEMESTRE 8

Département Informatique, Microélectronique, Automatique

Polytech'Lille, Villeneuve d'Ascq

ROBOT ASSISTANT DANS UNE MAISON INTELLIGENTE



Déroulement complète du projet :

http://projets-imasc.plil.net/mediawiki/index.php?title=Maison_intelligente

Réalisé par :

Josué RUKATA MAROY

Encadrant :

Romain TRIBOUT

Année Scolaire 2012-2013

I. INTRODUCTION

A l'école Polytech'Lille, chaque année, les étudiants de 4ème année IMA (Informatique Microélectronique Automatique) travaillent sur un projet tutoré tout au long du semestre 8. C'est dans ce cadre que j'ai travaillé sur le projet intitulé « Robot assistant dans une maison intelligente » proposé par le CITC-euraRFID, Centre d'Innovation des Technologie sans Contact.

Le projet consiste à développer un robot capable de suivre une personne dans une maison. Pour étudier la problématique d'un tel robot, le projet est restreint à l'usage du robot Robotino disponible à polytech-Lille. Ainsi, le résultat devrait permettre au CITC-EuraRFID d'imaginer les nouvelles fonctionnalités qu'ils pourront proposer dans la maison intelligente qu'il développe, en occurrence un robot capable de suivre une personne dans la maison, un robot capable de se localiser et de circuler dans la maison, ...

Le but de ce rapport ne consiste pas uniquement à vous présenter de manière exhaustive les aspects techniques développés dans le projet, il me permettra également de vous exposer les difficultés auxquels j'ai été confronté durant la réalisation de ce projet qui était réduit au simple fait que le robot Robotino devrait être capable de suivre une personne.

En premier lieu, je vous présenterai succinctement l'objectif du projet. Ensuite, je parlerai en détail du travail réalisé en passant par l'étude du projet, pour finir par les problèmes rencontrés et les solutions proposées.

II. RAPPORT D'ACTIVITE

1. Expression de besoin

Le CIRC-auraRFID, Centre d'Innovation des Technologie sans Contact, est une association hébergée chez AuraTechnologie et qui a pour objectif de favoriser l'appropriation des technologies émergentes en anticipant les évolutions technologiques et sociétales et de favoriser les synergies entre les entreprises, les organismes de recherche et de formation. Leurs travaux permettent de développer un écosystème technologique et ainsi faire de la région Nord-Pas de Calais une référence européenne dans le domaine de l'Internet des Objets.

C'est dans ce cadre que le CIRC-AuraRFID développe un projet nommé Smart Home (Maison Intelligente) où les objets de la maison (portes, fenêtres, frigo, ...) sont en interaction permanente avec les utilisateurs de la maison. Ces interactions sont développées par les technologies sans contact telles que le NFC, RFID, vision, laser, ultrason, Zigbee, capteurs sans contact, ...

Pour aller plus loin, le CIRC-auraRFID veut développer dans la maison intelligente un robot assistant capable de :

- se localiser et de circuler dans la maison,
- suivre une personne dans la maison
- répondre à l'appel d'une personne en dehors de la maison pour aider à décharger les cours
- d'atteindre un objectif donné (par exemple, déposer telle chose sur la table de la cuisine)

Le projet ayant plusieurs points à réaliser, il m'est proposé de travailler sur le deuxième : **réaliser un robot pouvant suivre une personne.**

Pour étudier la problématique d'un tel robot, le projet est restreint à l'usage du robot Robotino disponible à polytech-Lille.

2. Analyse du besoin

Le robot doit être capable de suivre une personne. Pour ce faire, il faut que le robot soit déjà capable de reconnaître son objectif, la personne. Le robot Robotino étant muni d'une caméra, nous allons nous servir des notions de traitement d'image existantes pour détecter les formes d'objet ou les couleurs qui pourront servir d'objectif au robot. Cependant, étant donné que les fonctions de traitement d'image sont gourmandes en espace mémoire et en temps de calcul, il est important de se rabattre aux performances du robot pour avoir une idée sur la taille des images à traiter et le temps de calcul afin d'avoir une bonne cohérence entre le traitement d'image et le déplacement du robot.

Caractéristiques importantes du Robotino :

- Un processeur PC 104+ cadencée à 500 MHz
- SDRAM 128 MB
- Carte mémoire Compact Flash 1Go
- OS Linux temps réel : Linux UBUNTU 9.1
- 2 ports USB 2.0



L'objectif à suivre sera mobile, il est alors important d'exploiter au mieux les limites du Robotino pour optimiser la sensibilité à détecter le déplacement de l'objectif.

3. Analyse et développement du projet

3.1. Traitement d'image

Pour réaliser la détection de notre objectif, nous allons utiliser les outils de traitement d'image déjà existants, en occurrence OpenCV qui est une bibliothèque graphique libre spécialisée dans le traitement d'images en temps réel.

Globalement, avec Opencv nous pouvons utiliser deux types de méthodes pour traquer des objets. Celles qui repèrent des détails caractéristiques de l'objet et celles qui se basent sur la détection de couleur. Combiner les méthodes serait une chose très intéressante car ça permettrait de détecter d'abord une personne par sa forme ensuite identifier la personne à suivre à partir d'une couleur particulière de ses habits par exemple. Cependant, la première méthode demande beaucoup de temps de calcul pour déterminer le contour global de l'objet.

Pour alléger nos calculs, dans la suite du projet, nous n'utiliserons que la deuxième méthode : traquer un objet qui possède une couleur donnée comme consigne.

Dans ce cas, nous allons nous débrouiller pour choisir une couleur rarement utilisée afin d'éviter que notre programme capte la même couleur se trouvant sur des objets indésirable.

Pour traquer notre couleur dans le flux vidéo, nous allons utiliser une technique appelée « *Binarisation* ». Elle consiste à séparer les pixels de l'image en deux classes distinctes par deux couleurs : blanc et noir. Le blanc correspondra à notre couleur à isoler, et le noir à toutes les autres couleurs. Ainsi, si notre couleur sélectionnée est le rouge, nous verrons sur notre image binarisée tous les éléments rouges apparaître en blanc sur un fond noir.



La couleur traquée est le jaune

Avant de faire la binarisation, il est important de tenir compte d'un point: les images capturées sont codées en RGB. Ce dernier est une façon de coder une couleur tel que interprété par notre système cérébral, il ne correspond pas donc aux couleurs du système physique réel. Il est alors difficile de réaliser des algorithmes pour faire la différence entre deux images en RGB. Nous allons commencer par convertir notre image RGB en image HSV (Hue, Saturation, Value, en français : TSV – Teinte, Saturation, Valeur) pour résoudre ce problème. Nous passerons l'image en HSV car ainsi nous pourrons nous baser sur la teinte et la saturation de la couleur. Pour binariser notre image originale, nous allons créer une nouvelle image masque de même taille mais sur un seul canal (noir blanc). Après, nous allons donc rechercher les pixels qui se situent dans un intervalle autour de la couleur de consigne, nous utiliserons une valeur de tolérance et elle sera la même pour la teinte et la saturation. Nous chercherons donc tous

les pixels qui vérifient :

$$H - \text{tolérance} \leq H_{\text{pixel}} < H + \text{tolérance} \ \&\& \ S - \text{tolérance} \leq S_{\text{pixel}} < S + \text{tolérance}$$

Sur OpenCV, la fonction *cvInRangeS* nous permettra d'appliquer cette opération directement à tous les pixels de notre image HSV et mettra à jour notre masque. Si l'expression est vérifiée pour un pixel donné de l'image, elle met le pixel correspondant du masque à 255 (blanc), sinon, elle le met à 0 (noir).

A ce niveau, il est possible d'isoler la couleur que l'on cherche. Cependant, si la couleur n'est pas bien choisie, il est possible d'observer des pixels indésirables éparpillés sur le masque. Pour s'en débarrasser, il faut utiliser certaines opérations appelées « *Opérations morphologiques* » parmi lesquelles on a une **érosion** et une **dilatation**.

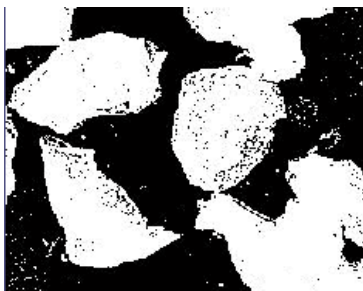


Image source



Image source + érosion



Image source + dilatation

L'érosion permet de supprimer les pixels isolés tandis que la dilatation permet de renforcer les groupes denses de pixels (dans notre objet). L'idéal serait d'utiliser les deux opérations pour bien filtrer l'image. Cependant pour de raison de temps de calcul, nous n'utiliserons que la dilatation pour avoir une bonne concentration du blanc sur notre objet cible, et nous nous arrangeront de trouver un objet ayant une couleur très particulière afin de diminuer le taux des pixels éparpillés (ça nous permettra de ne pas utiliser l'érosion).

Après avoir extrait notre cible de l'image, nous pouvons trouver le centre

de gravité de l'objet en calculant la moyenne des coordonnées des pixels blancs du masque.

En analysant le déplacement du centre de gravité qui représente notre objet sur les images binarisées, nous pouvons alors réaliser des algorithmes pour commander le déplacement du robotino.

3.2. Intégration du module de traitement d'image sur Robotino.

Robotino est en soi un mini-ordinateur sur lequel est installé le système d'exploitation Ubuntu 9.1. Le module de traitement d'image réalisé y est intégré normalement comme sur un PC.

La librairie RobotinoAPI2 possède des fonctions toutes faites pour déplacer le robot en C. Voici le code constituant l'algorithme du programme principal :

```
void drive(float * vX, float * vY, float * omega)
{
    while( Com_isConnected( com ) && FALSE == Bumper_value( bumper ) )
    {
        // fonction de traitement d'image
        detecteObjecte();

        // fonction de
        //calcul de la nouvelle direction que doit prendre le robot
        changeDir();

        // fonction pour commander le robot
        OmniDrive_setVelocity(omniDrive, *vX, *vY, *omega );
    }
}
```

III. Difficultés rencontrées

Vous aurez remarqué que j'ai beaucoup détaillé la réalisation de la détection de l'objectif que celui du déplacement du robot. Ceci parce que non seulement il m'a fallu beaucoup de connaissance en traitement d'image pour réaliser le projet mais aussi parce c'est la partie qui m'a prise beaucoup de temps à programmer.

Cependant, à l'intégration de la solution sur robotino j'ai été confronté au problème lié au déplacement à temps réel

En effet, le processus de traitement d'image n'était pas à temps réel. Ceci signifie que le processus ne répondait pas dans l'intervalle de temps correct pour envoyer les commandes de déplacement au robot. Au début, sur robotino, le programme de traitement d'image prenait deux secondes pour traiter une image et envoyé la commande de déplacement. Après quelques efforts d'optimisation dans les codes, ce temps de calcul fut réduit à 1 seconde mais cela n'aura pas changé grand-chose sur la capacité du robot à suivre cible. Finalement le processeur du robotino cadencé à 500Mhz ne nous a pas beaucoup aidés pour réaliser ce que nous voulions.

CONCLUSION

Sur le plan scolaire, le projet était très intéressant et très formateur. Pour traiter le problème, je suis passé par le traitement d'image avec OpenCv qui m'a permis de découvrir les différents algorithmes utilisés dans la détection d'objet en partant d'une simple image plate capturé par une caméra.

J'ai été confronté aux problèmes liés au limite matériel, des problèmes sur lesquels je n'avais jamais été confronté mais qui m'ont permis de prendre conscience du quotidien d'un ingénieur qui n'est pas limité à donner des résultats aux problèmes mais aussi d'être capable de comprendre pourquoi sa solution ne fonctionne pas.

L'objectif du projet n'était pas de fournir un résultat pédagogique mais plutôt un résultat exploitable par industriel. A ce point, j'ai failli à ma mission car le résultat final n'était pas exploitable avec le programme réalisé en C. Pour essayer de sauver le projet, j'ai essayé de réaliser ce même sujet en utilisant RobotinoView. Ce dernier présentait l'avantage de faire le traitement d'image sur le PC où il était installé. De ce fait, le robotino était déchargé des opérations de traitement d'image et je pouvais observer une amélioration énorme sur la sensibilité du robot à suivre son objectif. Mon projet scolaire s'est terminé à ce point, cependant, le CITC continue à travailler sur la réalisation d'un tel robot, un robot capable de suivre une personne.