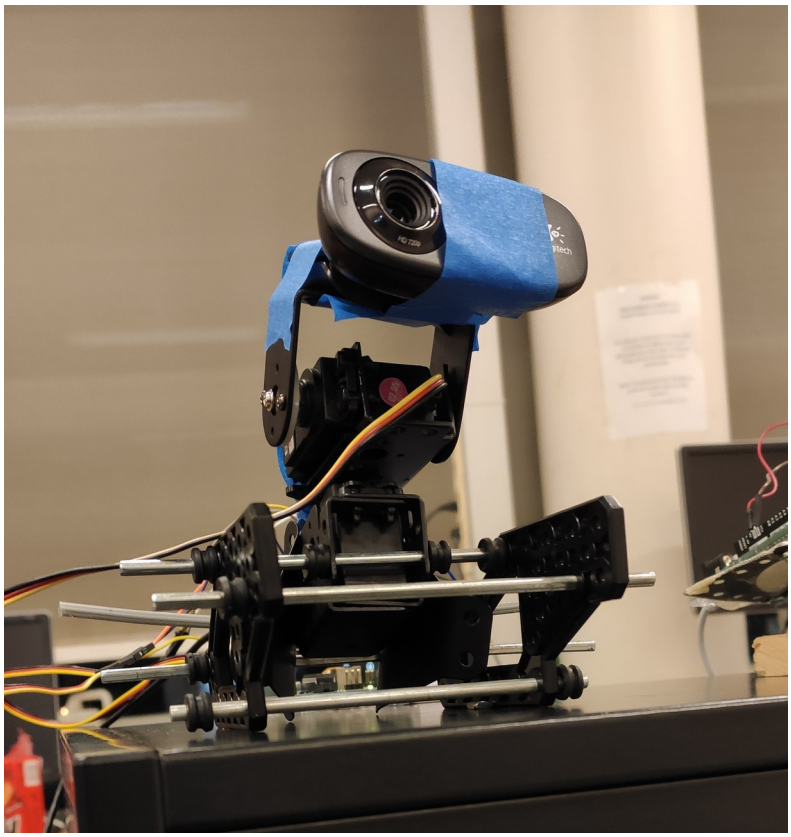
CAMARA Almamy
CHOLLET Alexis

Rapport d'introduction à l'informatique embarqué :Trac king d'un objet rouge



07/12/19

Sommaire

I Introduction

L'objectif du projet de robotique est de construire un robot qui sera en mesure de suivre un objet de couleur, qu'il soit fixe ou en mouvement. Pour cela on pourra agir sur deux axes du robot, le PAN et le TILT. Pour réaliser ce projet, nous disposerons de deux servo-moteur d'un arduino, d'une caméra, ainsi que de divers matériaux utiles pour le montage d'un robot.

II Description de la solution

Communication

Nous avons commencé par connecté une LED afin de pouvoir rendre explicite la communication entre l'arduino et l'ordinateur par la liaison série RS232 via l'IDE arduino. Ensuite nous avons attaché les deux servo-moteur à l'arduino au niveau des patte 6 et 7 et qui réagissent en fonction de ceux qui est écrit dans le fichier /dev/ttyACM0. En effet saisir 'q' (respectivement 'd') permet d'agir sur l'axe PAN du robot en le faisant pivoter par la droite (respectivement par la gauche). De même, saisir 'z' (respectivement 's') permet d'agir sur l'axe TILT en faisant pivoter le robot de haut en bas ou de bas en haut.

Détection de couleur + implémentation du tracking

On a choisi de détecter le rouge, car il s'agit de la couleur la moins présente dans notre environnemnt expérimentale. Pour cela on va parcourir l'ensemble des images produites par la caméra pixel par pixel, à la manière d'un tableau. On considérera que le pixel traité esr rouge si si la composante rouge a une valeur au moins deux fois supérieur à celle de la composante verte. Dans ce cas, on stocke la position des pixels rouge détecté dans les variables x et y (**x = count%320 et y = count/320 où count la variable de parcours du 'tableau'**). Le nombre de pixel rouge ainsi que leur position est enregistré pour ensuite calculer le barycentre de l'objet rouge détecté. Ensuite on calcule la différence en abscisse et en ordonnée entre le barycentre de l'objet rouge et le centre de l'image :

- erreurX = xmilieu - averagePosx;
- erreurY = ymilieu - averagePosy;

le signe indique la position de l'objet détecté par rapport au centre de l'image :

- Si erreurX< valeurSeuilX :
  - L'objet est trop à gauche par rapport au centre de l'image : on devra écrire 'd' dans le fichier ttyACM0 pour repositionner le robot pour recentrer l'objet au centre de l'image.
- Si erreurX> (-valeurSeuilX) :
  - L'objet est trop à droite par rapport au centre de l'image : on devra écrire 'q' dans le fichier ttyACM0 pour repositionner le robot pour recentrer l'objet au centre de l'image.
- Si erreurY> valeurSeuilY :

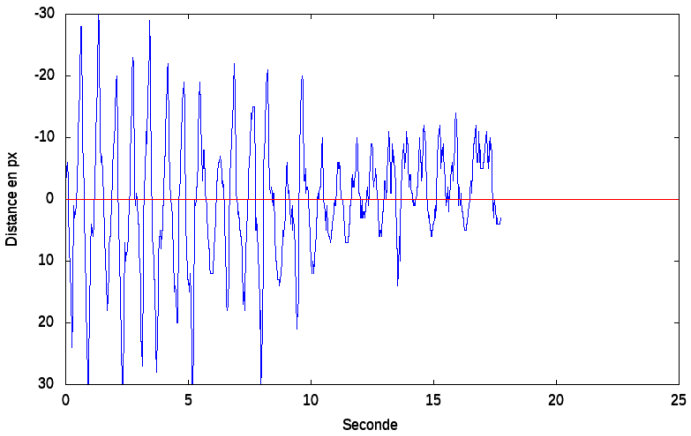
- L’objet est trop en bas par rapport au centre de l’image : on devra écrire ‘z’ dans le fichier ttyACM0 pour repositionner le robot pour recentrer l’objet au centre de l’image.

- Si erreurY<(-valeurSeuilY) :

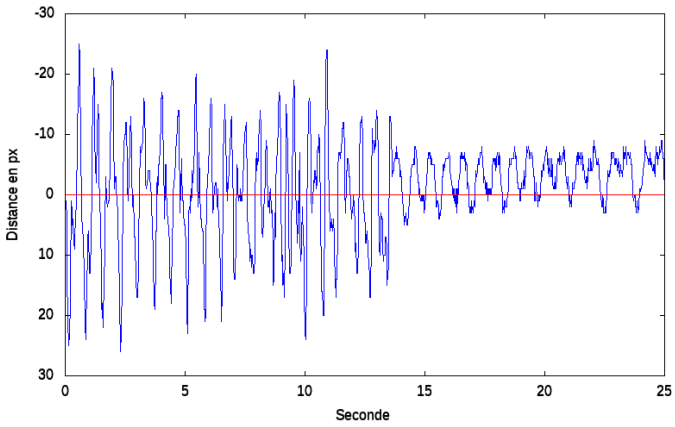
  - L’objet est trop en haut par rapport au centre de l’image : on devra écrire ‘s’ dans le fichier ttyACM0 pour repositionner le robot pour recentrer l’objet au centre de l’image.

Les variables valeurSeuilX et valeurSeuilY ont été déterminé au cours de plusieurs expérience au cours desquelles on a fais suivre un objet rouge que l’on fais osciller au cours du temps.

Ci-dessous les graphiques obtenu en faisant varier les paramètres de valeurSeuilX et valeurSeuilY :



Graphe 1: Erreur relative sur l'axe des abcisses avec valeurSeuilX=15, valeurSeuilY=15; l'objet est considéré comme étant au centre de l'image si l'erreur est compris entre 10 et -10



Graphe 2: Erreur relative sur l'axe des abcisses avec valeurSeuilX=10, valeurSeuilY=10; l'objet est considéré comme étant au centre de l'image si l'erreur est compris entre 6 et -6

On peut voir que le robot effectue plus efficacement le tracking de l’objet en rouge avec les paramètres qu’on fournit dans la seconde expérience que dans la première. Cependant, on peut observer du bruit sur les graphes, même quand l’objet est immobile. Cela peut signifier que soit notre code est perfectible et/ou que l’on a un défaut matériel ou de montage au niveau des servo-moteur.

III Conclusion

En résumé, on peut donc dire que je n’ai pu atteindre mon objectif qu’à moitié ; si le robot s’est montré capable de suivre tout objet qui rentre dans son champ de vision, il est incapable de se stabiliser lorsque la cible est immobile, sans doute pour les raisons qui ont été invoqué précedemment. De plus, les paramètres de l’environnement, comme la luminosité variait d’un jour à l’autre.