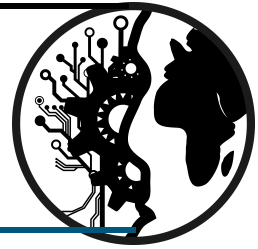


Projet

Contrôle basé vision d'un vecteur voiture

Laurent Beaudoin & Loïca Avanthey

Épita 2020



Avant propos

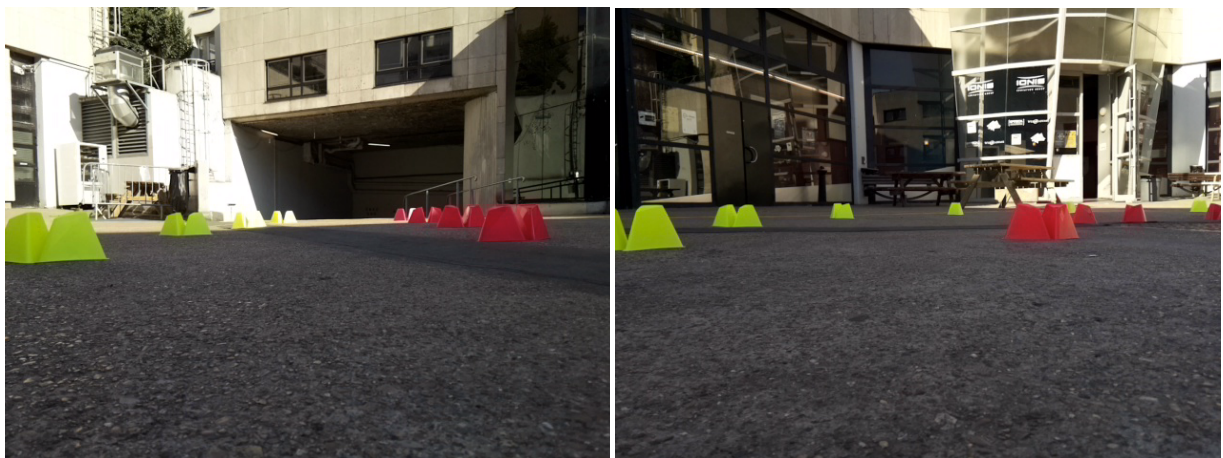
Avec le projet Télédétection, nous avons travaillé sur un algorithme embarqué à bord d'un satellite pour détecter les images d'intérêt. Dans ce projet, nous allons travailler à la mise au point d'un algorithme d'interprétation d'images en temps réel et qui est destiné à piloter une mini-voiture pour que celle-ci reste dans un parcours défini, ce qui revient à corriger automatiquement ses dérives par rapport à une trajectoire idéale.

Dans le cours Vision par Ordinateur, nous avons vu toutes les bases nécessaires pour utiliser une caméra, détecter et identifier des objets sur des images. Cela va nous servir pour asservir la voiture en position grâce à des plots colorés placés de part et d'autres de la trajectoire à réaliser (plots rouges à gauche et jaunes à droite).

Algorithme de vision par ordinateur

Le projet se fait par groupe de 4 personnes (les groupes que vous avez formé pour le projet cars de TELE).

Les circonstances exceptionnelles de cette année ne devrait pas nous permettre de tester vos algorithmes sur le vrai prototype dans l'immédiat. Aussi, nous allons utiliser des vidéos embraquées qui ont été enregistrées sur le prototype que l'on vous a donné dans le cadre du projet initial de télédétection et que l'on aurait du compléter dans le cadre du projet VOIR.



Deux *frames* extraites des vidéos

Dans le cadre de votre évaluation, vous aurez à rendre un mini-rapport (5 à 10 pages maximum, soyez bref et convaincant) et votre code source opencv. Cela vous sera précisé en détail dans l'exercice 4.

EXERCICE 1 (*Conception et réflexion*)



Écrivez un (ou des) algorithme(s) pour détecter automatiquement les plots rouges et jaunes sur les vidéos. Nous vous laissons le choix sur les méthodes à utiliser pour arriver à vos fins (il n'y a pas qu'une solution possible). Dans tous les cas (une ou plusieurs idées), décrivez bien le ou les algorithmes et faites en une analyse critique (justifiez chaque étape de votre solution, leurs points forts / points faibles attendus, définissez des critères d'évaluation de performance mesurables pour les valider comme le pourcentage de détection des cibles, les fausses détections, le temps de calculs etc.). Bref, mettez dans la partie conception et réflexion de votre rapport toutes les informations qui permettront de mettre en valeur cette première étape de votre travail.

EXERCICE 2 (*Implémentation de la détection*)



Codez votre ou vos algorithmes en openCV. Votre algorithme doit afficher sur les images du flux vidéo la position des plots détectés (marqueur en couleur par exemple) (1^{er} niveau de difficulté) et leur taille (via un cercle, une ellipse ou un rectangle par exemple autour des groupes de 2 plots de même couleur) (2^e niveau de difficulté).

Bonus : sur une des séquence, il y a un pigeon. Un bonus si vous le détectez automatiquement :).

EXERCICE 3 (*Réaction... virtuelle*)



Maintenant que vous avez identifié vos cible, il faut simuler la correction que vous devez effectuer pour rester au centre des plots (trajectoire idéale).

Complétez votre algorithme pour calculer le milieu entre les plots rouges et jaunes les plus proches de la voiture et afficher une flèche (ou assimilé) qui part de ce milieu et va jusqu'au centre de l'image. Si les plots proches de l'une des deux couleurs ne sont pas visibles, faire aller la flèche du centre jusqu'au bord de l'image. Afficher dans la console l'ordre à exécuter parmi les suivants (avant, arrière, stop, gauche ou droite) et son intensité (0 pour rien et 255 pour max).

EXERCICE 4 (*Rendu*)



Pour le rendu, vous nous fournirez le projet de votre code (fichiers sources + CMakeLists + README) dans une archive `aux_noms_de_votre_équipe.tar.gz` et un micro-rapport (5 à 10 pages maximum) qui contiendra ce qui vous est demandé à l'exercice 1 ainsi qu'une partie résultats et performances obtenues. Nous testerons votre programme sur une autre vidéo issue de la même série que celles que nous vous avons données.



Prêts ?