

Rapport récapitulatif

Eyes On Cars

20.05.2020

Date dernier rapport : 27 avril 2020
Date prochaine réunion : Fin de projet

Contexte :

Notre participation au projet touche à sa fin. Dans ce dernier rapport, nous récapitulerons les méthodes utilisées dans la reconnaissance des véhicules et les ouvertures potentielles vers des approches plus prometteuses. Nous présenterons aussi notre vision sur la mise à disposition des données.

Organisation spatiale

Nous avons ici choisi de n'utiliser qu'une seule caméra (dans un souci de prise en main du projet et de ses besoins) pour suivre le déplacement de tous nos véhicules. Nos tests de développement ont été réalisés avec une webcam logitech 720p avec un angle de 60 degrés. Nous plaçons la caméra au centre de la zone à observer et placée à une hauteur suffisante pour observer la totalité de la scène. Hauteur qui est déductible via un peu de trigonométrie. Nous avons noté que la caméra se trouve alors à une distance très lointaine pour un grand circuit ce qui, à l'avenir, peut poser problème si le projet évolue. La possibilité de disposer de plusieurs caméras est à considérer mais il faut savoir qu'elle imposera des contraintes sur le regroupement des flux pour ne pas identifier plusieurs fois le même véhicule sous différents angles.

Rappel sur le code

Rappelons ici que nous utilisons principalement le module OpenCV de python qui contient déjà un grand nombre de fonctions de haut niveau pour le traitement de l'image et dont nous ne ferons pas le détail dans ce rapport. La documentation très complète sur internet et les multiples applications sur des projets en tous genres nous ont permis de facilement prendre en main le module. Nous nous sommes considérablement appuyé sur le site Pymotion (<https://pymotion.com/>) qui présente de façon très détaillée et très pédagogue l'utilisation de ce module.

La classe ImageTreatment (*fichier Mother.py*)

Cette classe contient les modifications de base des images (non déjà présentes grâce à OpenCV) permettant notamment de seuiller une image et de la filtrer avec trois filtres au choix (Median, Gaussien et Moyen). Une fonction d'affichage des états progressifs est aussi disponible pour permettre à l'utilisateur d'observer l'évolution des traitements et leurs impact pour mieux extraire les formes dans la prochaine étape.

La classe ImageToForm (*fichier ImageToForm.py*)

Cette classe contient les méthodes plus poussées de traitement d'image et est fille de la classe précédente. Elle permet notamment de se plonger dans une autre dimension, le HSV, qui est mieux adapté pour le traitement d'image. Plusieurs méthodes d'extraction de formes sont aussi mises à disposition et nous en ferons le détail dans la section **Méthodes de reconnaissance**. C'est aussi dans cette classe que se trouve l'identification des véhicules et la localisation dans l'image.

Le main (*fichier main.py*)

Ce fichier python contient le code principal qui permet de lancer l'identification des véhicules (cette étape peut prendre plusieurs secondes) puis le tracking de ceux identifiés. Un fichier CSV sera ouvert pour enregistrer les différentes données que l'utilisateur souhaite mettre à sa disposition. Une fois celui-ci lancé, si l'identification est mal faite ou que l'expérience est terminée, il faut appuyer sur la touche "Echap" pour quitter le flux caméra et terminer le programme.

Activités réalisées durant le semestre

Soustraction de fond

La soustraction de fond nous a paru une première bonne idée. Cependant pour la mettre en place il aurait fallu obtenir une image du fond avant l'introduction des véhicules pour ensuite pouvoir réaliser la soustraction. Cette méthode est très peu flexible au changement de décor qui nécessite constamment de réinitialiser l'image servant de fond. Elle est aussi très vulnérable aux jeux d'ombres et de lumières. Nous avons donc fini par abandonner cette approche.

Seuillage et Filtrage

Une première idée dans le domaine des niveaux de gris était de seuiller les images selon un certain niveau car les couleurs des véhicules étaient en fort contraste avec le fond. Cependant, cette méthode s'est avérée infructueuse malgré les différents filtres appliqués car elle est dépendante de la couleur du fond et est, comme la précédente, sensible aux jeux d'ombres et de lumières. Ces méthodes nous seront cependant utiles mais seulement après avoir réalisé plusieurs étapes préliminaires.

Domaine HSV

Le domaine HSV (du français Teinte Saturation Valeur) est une nouvelle interprétation plus perceptive des couleurs. Nous avons décidé d'utiliser cette approche pour mieux capturer les véhicules et éviter notamment l'impact de la luminosité dans la détection. Même si des améliorations sur la reconnaissance ont pu être observées, l'approche faite reste encore trop sensible à la luminosité. Une combinaison linéaire des plans H, S et V est parfois nécessaire pour obtenir un bon moyen de reconnaissance. A ce jour, nous n'avons pas trouvé de combinaison plus performante que les autres qui sont encore trop dépendantes de l'environnement du circuit (ie le fond utilisé).

Identification des véhicules

Après avoir reconnu les formes représentant des véhicules, il était important de pouvoir les identifier pour les dissocier entre eux. Nous avons initialisé deux approches qui se sont révélées toutes les deux infructueuses. La première consistait à étiqueter chaque véhicule avec un Qr Code qui pourrait être lu par la caméra puis traité avec le module OpenCV pour en tirer les informations des différents véhicules. Le circuit utilisé étant trop grand, la distance au sol de la caméra ne permet pas une bonne lecture des Qr Code. La seconde méthode était d'identifier les véhicules grâce à un court texte (ou une plaque d'immatriculation) écrit sur le toit des véhicules en utilisant le module PyTesseract. Cependant, comme la précédente méthode, la grande distance séparant la caméra du sol et la forte concentration de bruit en fond empêchent la reconnaissance de ces identificateurs. Le problème d'identification est donc toujours en attente d'être résolu mais nous pensons que la piste des réseaux de neurones est à suivre.

Tracking des véhicules

Le tracking s'effectue à l'aide de la classe MultiTracker de OpenCV. C'est une méthode de haut niveau qui permet de suivre un objet précédemment identifié. Un rectangle sera alors mis à jour sur le flux caméra avec les mouvements des objets reconnus. Il est important de relever que la collision de deux objets, notamment ici des véhicules, peut entraîner une perte du focus de l'algorithme. Deux fenêtres de suivi peuvent donc se superposer dans le cadre d'un rapprochement et l'un des véhicules ne sera donc plus traqué. Bien que compliquées à mettre en place, des solutions pour remédier à ce problème peuvent être implémentées dans le main.

Mise à disposition des données

La mise à disposition des données est faite dans un CSV où nous mettons à disposition la position sur les axes X et Y ainsi que la vitesse sur les

axes X et Y (celles-ci sont converties en cm/s pour être plus facilement manipulables). Elles sont remplies séquentiellement dans le CSV sans que l'historique soit supprimé mais on peut imaginer que les CSV utilisés ne contiendront que l'information pertinente (ie les positions et vitesses courantes des véhicules). C'est, à l'avenir, la solution la plus viable et performante. Dans notre interprétation du projet, nous avons prévu d'utiliser un fichier CSV par véhicule de sorte que les informations soient plus facilement accessibles. Le main contient dans ses premières lignes des valeurs en dur qui devront être changées lors du test de l'application.

Pistes d'améliorations

Nous conseillons toujours de s'inspirer du site Pymotion qui répond aux mêmes problématiques que celles auxquelles nous faisons face. Cependant, si nous avons, dans un premier temps, écarté l'idée d'utiliser l'intelligence artificielle dans la reconnaissance des véhicules, elle semble de plus en plus pertinente. Le framework Tensorflow allié à OpenCV pourrait permettre une véritable amélioration dans la détection initiale et l'identification des véhicules sur le flux vidéo.

Conclusions sur le projet

Malgré nos efforts, plusieurs améliorations sont encore à proposer avant de pouvoir obtenir une reconnaissance et identification robuste des véhicules. Le traitement d'image seul semble être insuffisant pour identifier les voitures et nous conseillons donc de se tourner vers les réseaux de neurones à convolution (CNN) très utilisés dans ce domaine. L'utilisation de plusieurs caméras semble aussi être inévitable si le circuit devient trop grand.