Tracking de voiture

Sommaire

Intro

Le tracking d'objet par masque de couleur n'est pas la meilleure solution car l'objet à traquer n'est pas forcémment d'un couleur différente de son décors. Cependant c'est une approche simple, et ça rentre dans le cadre du programme !

Installer OpenCV

Python

Pour pouvoir utiliser la dernière version d'OpenCV, il faut avoir au minimum **Python 3.5**.

> Note : Je n'ai pas réussi à trouver la version de Python que l'on aura le jour du concours. Je pense que ça sera celle installée sur les PC où on sera.

On peut se servir d'anciennes versions d'OpenCV, sur d'anciennes versions de Python, mais l'installation devient assez compliquée. De plus, il n'est jamais conseillé de programmer avec des versions antérieures. Dans le monde de l'OpenSource s'il y a des mises à jour, ce n'est pas pour faire payer l'utilisateur une nouvelle fois, il y a de réelles raisons de sécurtié, d'optimisation, et d'évoultion. 😄

Installation

Ensuite il suffit d'installer OpenCV comme tout autre librairie.

```
pip3 install opencv-python
```

Remarques de base sur la programmation moderne

Markdown

Le markdown est un language de texte adapté à la programmation. En science, il y a LaTeX, en programmation c'est le markdown. C'est un peu comme de l'HTML, il permet de mettre en forme du texte explicatif lié à la programmation.

Tout sujet de TP devrait, selon moi, être écrit en Markdown 😉.

Il est simple d'utilisation. Ce fichier de présentation est écrit en markdown. Ouvrez `README.md` dans les fichiers ci dessus pour voir à quoi ca ressemble.

Github

Github, c'est un peu comme Google Drive, mais pour la programmation Open Source. C'est un site où on peut sauvegarder tout ses codes, et les partager. C'est un complément du logiciel git.

Il a de nombreuses fonctionnalités, est très utile, et très utilisé ! Il pourrait être utilisé en cours d'informatique pour partager les fichiers utiles aux TP (notamment les sujets écrit en Markdown 😉). Il pourrait aussi être utilisé par les élèves pour ranger leurs codes et leurs DM d'informatique par exemple.

Un *repositry* ou repos, est un 'dossier' contenant un ensemble de fichiers. Par exemple sur mon compte j'ai, entre autre, un repos *Colloscope* pour l'application que l'on a faite, et un repos *Cours*, où je range tous les programmes faits en cours.

Sur github on ne peut que télécharger des repos en entier en cliquant sur le bouton 'Clone or download'. Si on veut télécharger juste un fichier, il faut faire copier coller de son code source (raw).

linter

Je travaille avec l'*IDE* Visual Studio Code (de loin le meilleur éditeur de texte pour coder à ce jours).

Il a l'avantage d'incorporer des linter. Les **linter** ce sont des programmes qui tournent en arrière plan même temps qu'on tape notre code, et qui soulignent nos fautes de programmation comme le ferait Word pour les fautes d'orthographes et de conjuguaisons.

Par exemple il souligne `np.linespace` car cette fonction n'existe pas (c'est `np.linspace`). Il soulinge les problèmes d'indentation sous python, ... il corrige beaucoup d'erreurs ! Cela prévient de passer du temps à déboguer un programme pour des erreurs bêtes.

> Je pense d'ailleurs qu'apprendre à programmer là-dessus peut être très avantageux. Le problème est que le jour du concours on est sur IDLE, et passer de VS Code à IDLE, c'est très compliqué !

importer OpenCV

Pour importer openCV dans un programme Python, il suffit de mettre :

```
import cv2
```

et de s'en servir comme ça :

```
video = cv2.VideoCapture("Ressources/test.mp4")
```

Cependant, j'utilise un linter qui va lire (sur mon PC, là où OpenCV est installé) toutes les fonctions existantes d'OpenCV pour vérifier ce que j'écris. Et y'a un petit bug à ce niveau là, c'est pourquoi je l'importe de cette manière :

```
import cv2.cv2
```

Mais les deux sont équivalents.

Modules

J'aime bien découper mon code en plusieurs fichiers différents. En python pour importer du code d'un fichier qui est dans un sous dossier, il faut le faire comme si c'était un module.

Sauf que mon linter ne reconnait les modules que s'il y a un fichier `__init__.py` dedans. Voilà pourquoi j'ai ce fichier inutile dans le dossier `src`.

Structure du programme

main

Le fichier `main.py` contient tous les paramètres variables du code (chemin des fichiers, couleurs, ...) c'est lui qu'il faut lancer et éditer pour se servir du tracking. Il se veut court et simple.

src

Comme dans bcp de structures de code, j'ai mis le coeur du programme dans un dossier source (`src`). J'ai découpé le code en deux pour donner deux exemples simples, mais tout aurait pu (dû) être condensé en une seule fonction.

media

Le dossier media quand à lui contient toutes les images et vidéos à analyser grâce au code. Le dossier '`media`' aurait lui-même pu se trouver dans le dossier '`src`' mais les fichiers qui s'y trouvent ne sont pas nécessaire au fonctionnement du programme.