Soustraction de l’image moyenne sur une vidéo

# Objectif

L’objectif de cette tâche est de faciliter la détection de la balle sur l’image en supprimant ce qui est superflu et pourrait gêner la détection de la balle.

# Hypothèses de travail

Le cadre de la vidéo est fixe.

Les images de la vidéo sont encodées en RGB.

# Méthodologie et code

La vidéo est traitée image par image. L’idée générale est de calculer l’image moyenne des images encadrant l’image traitée. La taille de l’intervalle est donnée par la variable tailleMoy. Par exemple si la taille de l’intervalle est de 9, on calcule l’image moyenne avec les 4 images précédentes, l’image que l’on souhaite traiter et les 4 images suivantes. Ensuite, on soustrait à l’image traitée cette image moyenne, ce qui permet de supprimer totalement ce qui est fixe sur la vidéo et d’atténuer ce qui bouge peu.   
*Par soustraction d’image on entend soustraire les valeurs de rouge, bleu et vert de chaque pixel de l’image.*

Dans la mesure où ce traitement ne nécessite que quelques images après l’image en cours de traitement, il peut être fait en temps réel.

Dans un premier on travaille sur une image, puis sur une suite d’image enregistrées sur un ordinateur. Ensuite nous avons transposé la démarche à une vidéo enregistrée et à une vidéo filmée depuis une caméra.

Ce programme nécessite les modules python suivants : os, numpy, PIL (module Image) et OpenCV (cv2).

## Travail sur des images

Une image contenant texte

Description générée automatiquementOn définit les deux fonctions suivantes qui reçoivent des listes d’images (de la classe Image du module PIL). La soustraction de l’image est faire à l’aide du module OpenCV qui nécessite une conversion des tableaux de RGB en BGR (<https://stackoverflow.com/questions/14134892/convert-image-from-pil-to-opencv-format> , 1ère réponse).

Ensuite, il ne reste plus qu’à ouvrir et stocker toutes les images dans une liste en envoyant dans la fonction subImageMoy une sous liste contenant l’image à traiter au centre.

Une image contenant texte

Description générée automatiquement

Le cas des images « en bord » de liste n’a pas été traité.

## Travail sur une vidéo enregistrée

Pour la lecture et l’écriture de vidéo, on utilise le module OpenCV. On définit dans un premier temps une fonction qui crée une liste d’images (plus précisément de tableaux représentant des images) à partir de l’adresse de la vidéo en prenant en paramètre le nombre d’images à sauvegarder par seconde (potentiellement différent de celui de la vidéo).

Une image contenant texte

Description générée automatiquement

Les deux fonctions précédentes sont redéfinies pour prendre en argument une liste d’array BGR et on crée également une fonction appliquant l’algorithme de soustraction sur une liste d’image (reprise du code précédent) :

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

*Annexe : pour permettre une visualisation, on a aussi créé une fonction créant une vidéo à partir de la liste d’images mais elle ne sera sans doute pas utilisée pour le tracking.*

Une image contenant texte

Description générée automatiquement

Le code principal de cette partie est le suivant :

Une image contenant texte

Description générée automatiquement

Quelques liens : <https://morioh.com/p/10d5b5a6f3d2>  
 <https://www.it-swarm-fr.com/fr/python/python-creation-de-video-partir-dimages-en-utilisant-opencv/831300377/>

## Travail sur une vidéo en temps réel

*Non fait*

# Résultats

## Travail sur des images

Une image contenant texte, moniteur, écran, sombre

Description générée automatiquementLa soustraction d’image en utilisant OpenCV fonctionne très bien puisque seul la balle et les joueurs (c'est-à-dire ce qui bouge). On s’est intéressé à l’influence de la variable tailleMoy. Qualitativement, on peut dire qu’augmenter tailleMoy permet de faire davantage ressortir les joueurs. En effet leur mouvement est plus lent que la balle, donc une moyenne sur un petit nombre d’images ne s’éloigne que très peu de l’image centrale et la soustraction efface alors les joueurs.

tailleMoy = 19

Une image contenant texte, sombre, ciel nocturne

Description générée automatiquement

tailleMoy = 9

## Travail sur une vidéo enregistrée

### Première vidéo

On a d’abord testé le programme sur cette vidéo [Table Tennis in Slow Motion.mp4](Table%20Tennis%20in%20Slow%20Motion.mp4)  
Le résultat obtenu est dans la vidéo suivante : <videoSub.mp4>

La vidéo étant ralentie, on n’a pas d’effet de trainée de la balle et on voit bien que la soustraction de l’image moyenne fonctionne très bien. La seconde vidéo vise à montrer que cela n‘est en réalité pas problématique.

### Deuxième vidéo

Cette vidéo [TT Real Speed.mp4](TT%20Real%20Speed.mp4) est un jeu en vitesse réel. On obtient les résultats suivants avec tailleMoy=9 : [SUB9 TT Real Speed.mp4](SUB9%20TT%20Real%20Speed.mp4) et tailleMoy=3 : [SUB3 TT Real Speed.mp4](SUB3%20TT%20Real%20Speed.mp4).

La vitesse de la balle n’a pas d’impact sur la soustraction de l’image moyenne. On retrouve cependant l’impact de la variable tailleMoy (« aura » autour des joueurs très visible sur la première vidéo).

Enfin il est primordial de noter que la table étant immobile, elle est effacée lors de la soustraction. La détection d’un rebond est donc plus difficile puisque l’on ne peut se baser que sur la trajectoire de la balle et pas sur un contact visible de la balle et de la table.

## Travail sur une vidéo en temps réel

*Non fait*