

# Vision par ordinateur

## Sujet: Comptage des passagers

Promotion 23

Redigé par:

Mamadou Ben H Cissoko & Abdoul-Fatao OUEDRAOGO

January 5, 2020

## 1 Introduction

De nos jours , la vision par ordinateur est un domaine scientifique interdisciplinaire qui traite de la façon dont les ordinateurs peuvent être conçus pour acquérir une compréhension de haut niveau à partir d'images ou de vidéos numériques . Du point de vue de l' ingénierie , il cherche à automatiser les tâches que le système visuel humain peut effectuer. La discipline scientifique de la vision par ordinateur s'intéresse à la théorie derrière les systèmes artificiels qui extraient des informations à partir d'images, dont les résultats sont plus en plus optimal.

En effet, beaucoup de domaines ne se passent plus de la vision par ordinateur car cela va beaucoup plus vite qu'avec une vision humaine qui se fatigue. Par exemple dans le domaine du transport de plus en plus d'Autorités Organisatrices de Transport( **AOT**) sont désireuses de mieux connaître et évaluer la fréquentation de leurs réseaux afin d'adapter leur offre de transport et de dimensionner l'utilisation de leurs véhicules de façon optimale. La mise en place d'un système de comptage automatique des passagers répond ainsi à cet objectif de meilleure connaissance des flux de voyageurs. Grâce à des mesures complètes et fiables, une AOT ou un exploitant pourront ainsi facilement obtenir des données représentatives de la charge de leurs véhicules/trains et de la fréquentation de leur réseau mais aussi l'un des avantages de la vision par ordinateur est dans le domaine de suivi et de détection des objets est devenu très important et dont la demande de résolution des problèmes ne cessent d'augmenter , par exemple : **Vidéos de surveillance** , **Système de reconnaissance des personnes** etc.

## 2 Différence entre Détection des objets et Suivi des Objets

En vision par ordinateur la détection d'objet et le suivi d'objet sont deux termes qui confient très souvent les personnes donc avant de commencer notre travail il est important d'éclaircir ces deux concepts:

- **La détection des objets** est une méthode permettant de détecter la présence d'une instance ou d'une classe d'objets dans une image numérique.

Un détecteur d'objets est également généralement plus coûteux en calcul, et donc plus lent, qu'un algorithme de suivi d'objets.

Exemples: d'algorithmes de détection d'objets incluent les cascades Haar, HOG + SVM linéaire et les détecteurs d'objets basés sur l'apprentissage profond tels que les R-CNN plus rapides, YOLO et les détecteurs à un coup (SSD).

- **Le suivi des objets** consiste à prendre un ensemble initial de détections d'objets, à créer un ID unique pour chacune des détections initiales, puis à suivre chacun des objets lorsqu'ils se déplacent dans les images d'une vidéo, en conservant l'attribution d'ID.

Exemples : Dense Optical flow, Sparse optical flow, Meanshift and Camshift, Multiple object track finding algorithms et **Kalman Filtering** qui est un algorithme de traitement du signal très populaire utilisé pour prédire l'emplacement d'un objet en mouvement sur la base des informations de mouvement antérieures. L'une des premières applications de cet algorithme était le guidage de missiles! Également

comme mentionné ici, «l'ordinateur de bord qui a guidé la descente du module lunaire d'Apollo 11 vers la lune avait un filtre de Kalman».

### 3 Présentation de l'application:

Le programme est implémenté pour détecter le mouvement des personnes dans une vidéo et de compter aussi le nombre des personnes franchissant la ligne considérée comme ligne d'intersection qui coupe la vidéo en deux parties et il prend en entrée un seul paramètre. Il s'agit du nom de la vidéo à analyser. Il renvoie en sortie l'image correspondante à l'arrière-plan de la vidéo, les séquences d'images constituant la vidéo originale et les séquences d'images présentant le mouvement des objets. Pour exécuter notre projet :

**Étape 1 :** Il faut installer les librairies nécessaires; et ces librairies sont énumérés dans le fichier *requirement.txt* du projet.

**Étape 2 :** A partir du terminal ,tapez `python comptage-passagers.py -i` suivi du répertoire des vidéos de test. Exemple : `python comptage-passagers.py -i videos/example-02.mp4` et le résultat de sortie est le suivant dont on parlera de la procédure et l'implémentation

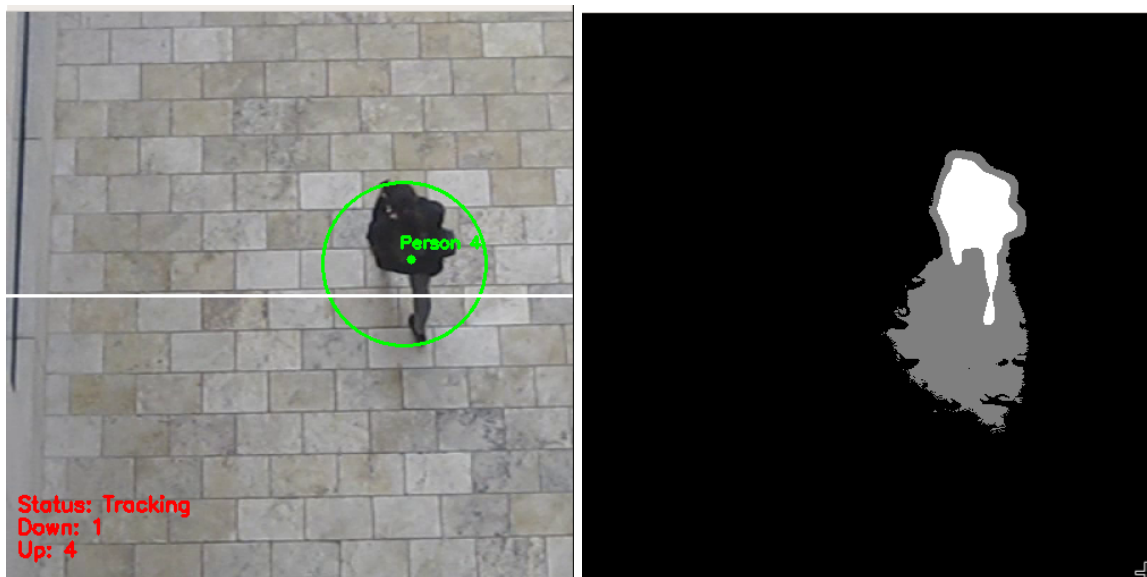


Figure 1: Comptage des personnes - Suivi de la personne(objet)

## 4 Principes de fonctionnement

Le projet sera divisé en deux parties : La première phase consistera à la phase de tracking et la deuxième phase celle de suivie et à laquelle on comptera les personnes.

Il existe plusieurs algorithmes dans opencv qui nous permet de détecter les objets dans les frames (Vidéos), comme par exemple **Frame differencing**, **Optical Flow** et **Background subtraction** Dans notre cas nous allons **background subtraction mog2** dans Opencv.

### 4.1 Étape de détection :

La première étape pour la soustraction est la modélisation d'arrière-plan et elle doit être suffisamment sensible pour reconnaître les objets en mouvement. La modélisation en arrière-plan doit donner le modèle de référence. Ce modèle de référence est utilisé dans la soustraction d'arrière-plan dans laquelle chaque séquence vidéo est comparée par rapport au modèle de référence pour déterminer une variation possible. Actuellement, filtre moyen et filtre médian sont largement utilisés pour réaliser la modélisation d'arrière-plan.

La méthode de soustraction de fond consiste à utiliser la méthode de la différence de l'image actuelle et l'image d'arrière-plan pour détecter les objets en mouvement, avec un algorithme simple, mais très sensi-

ble aux changements dans l'environnement extérieur et à une faible capacité anti-interférence. Cependant, il peut fournir le plus complet des informations sur l'objet dans l'arrière-plan.

Cette étape nous permettra de traiter toutes les frames de la vidéo avec les techniques de traitement d'images avec comme objectif d'avoir une détection uniquement des personnes une fois les différentes séquences d'images passées à notre réseau (modèle) pré-entraîné **cv2.dnn.readNetFromCaffe**.

- **Extraction de l'arrière-plan:**

Dans cette partie, nous avons défini un nombre des images  $N_{Frames} = 30$  de la vidéo pour construire l'image d'arrière-plan car nous avons remarqué qu'il n'était pas toujours nécessaire de considérer la vidéo entière pour obtenir de bons résultats et cela nous permet également de réduire le temps de calcul nécessaire.

A chaque séquence d'image nous détectons les différents objets à l'intérieur tout en calculant les coordonnées de l'objet qui nous permettra de calculer le centre pour qu'il puisse être suivi tout au long de la séquence.

Pour y parvenir, nous appliquons quelques techniques de traitement d'images dans le but de faciliter la détection des objets. Le filtre « **GaussianBlur** » a été appliquée dans le but de lisser l'image d'arrière-plan et l'image courante de la vidéo car elle a pour objectif de réduire en amont les sources de bruits dans l'image caractérisant le mouvement. La taille de la matrice a été fixée à **21x21**.

- Le filtre Canny :

Le filtre Canny est alors appliquée sur les images pour détecter les bordures des différents objets dans la scène avec une valeur de seuil 100, 190 cette valeur n'étant pas optimale avec d'autres valeurs la détection peut être beaucoup meilleure.

#### 4.1.1 Expérimentations

Dans cette section, nous avons conduit quelques expérimentations visant à observer la capacité de notre programme à détecter des mouvements dans une vidéo. La base de test utilisée est composée de vidéos provenant de youtube.

Nos premières expérimentations ont portées sur l'extraction de l'arrière plan des vidéos mais aussi de la détection des personnes présentes dans ces images et les images ci- dessous montrent des images d'arrière-plan obtenues en considérant différents nombres de séquences.



Figure 2: **Detection des personnes**

Nous voyons dans ces images qu'on arrive à détecter les personnes uniquement et tous les objets moins intéressants ont été ignorés cela est dû au fait qu'on a choisi un seuil pour lequel tout objet dont le son contour n'est supérieur à cette valeur doit être ignoré dans la fonction **cv2.contourArea** opencv. Dans cette image nous notons que la détection réalisée est quasi parfaite, seul les objets en mouvement sont détectés. le cercle encadre les personnages dans la vidéo originale.

Nous concluons que le nombre de séquences considérées influent nettement sur la qualité de l'extraction des arrière-plan. Cependant il n'est pas nécessaire de traiter toutes les séquences de la vidéo pour en extraire l'arrière-plan et effectuer la détection des objets.

## 5 Étape de Suivi du mouvement et Comptage:

A la phase de «suivi», pour chacun de nos objets détectés, nous créons un objet tracker pour suivre l'objet pendant qu'il se déplace autour du cadre. Notre objet tracker devrait être plus rapide et plus efficace que le détecteur d'objets. Nous continuerons de suivre l'objet jusqu'à ce que nous ayons atteint le temps de max disparition qui a été fixé à 40 , puis ré exécutons notre détecteur d'objets. L'ensemble du processus se répète ensuite.

Comme nous avons énuméré à l'étape de détection nous utiliserons un modèle pré-entraîné pour le suivi des personnes en mouvement. Cet modèle permet de détecter plusieurs objets mais comme nous on veut seulement détecter les personnes il faut alors choisir un bon seuil de confiance ce qui permettra une détection précise.

Afin de suivre et de compter un objet dans un flux vidéo, nous avons besoin d'un moyen de stocker des informations concernant l'objet , notamment:

- l'ID de l'objet
- les centroides precedent de l'objet(pour permettre de calculer facilement la direction dans laquelle l'objet se déplace)
- De savoir si l'objet a déjà été compté ou non

### 5.1 Suivie des personnes

Notre programme permet de suivre les objets en mouvement dans une vidéo. Pour suivre un objet avec notre modèle, nous suivons le centre de gravité de la boîte englobante qui entoure l'objet.

#### 5.1.1 L'algorithme de suivi des centroïdes

L'algorithme de suivi de centroïde est un processus en plusieurs étapes.

- **Étape 1: Détecter les objets et calculez les centroïde**

Grâce a la phase de détection nous avons pu détecté les differents objets et a partir de la nous avons calculé les coordonnées de boîte englobante (x, y) pour chaque objet détecté dans chaque séquence de frame de notre avec un identifiant unique associé a chaque objet.

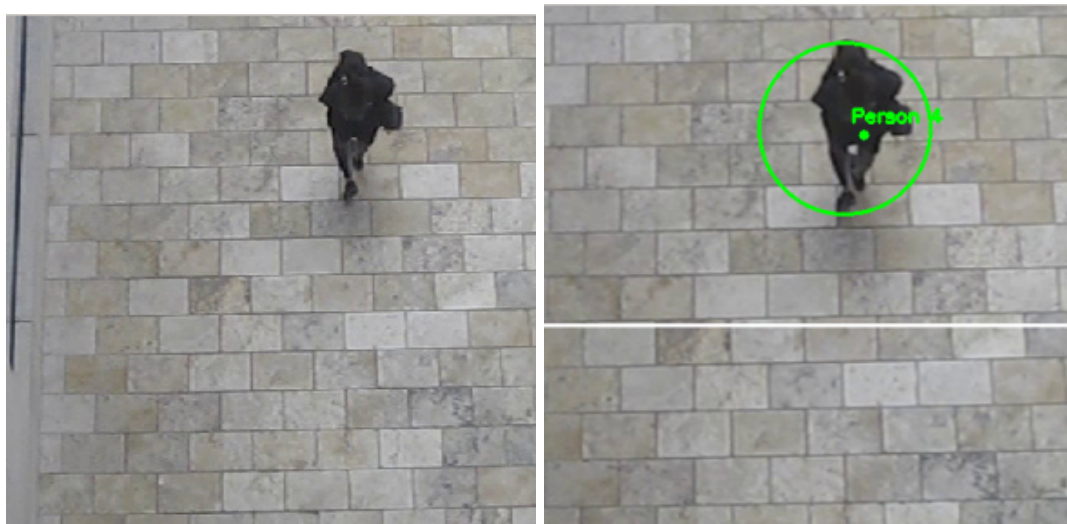


Figure 3: calcul du centre de gravité

L'image ci-dessus reflète le résultat de notre expérimentation car on peut clairement voir le centre de gravité à l'intérieur de la boîte englobante ce qui caractérise l'objet avec un identifiant unique.

- **Étape 2: Calculer la distance euclidienne entre les nouvelles boîtes englobantes et les objets existants**

Pour chaque image suivante de notre flux vidéo, nous appliquons l'étape 1 du calcul des centres de gravité des objets; cependant, au lieu d'attribuer un nouvel ID unique à chaque objet détecté (ce qui irait à l'encontre du but du suivi des objets), nous devons d'abord déterminer si nous devons associer des nouveaux centroïde d'objet (jaune) aux anciens centroïde d'objet ou bien si ce sont des nouveaux objets qui devons avoir leurs centroïde. Pour accomplir ce processus, nous calculons la distance euclidienne existants et les centroïde d'objet d'entrée. L'hypothèse principale de l'algorithme de suivi des centroïdes est qu'un objet donné se déplacera potentiellement entre les images suivantes, mais la distance entre les centroïdes pour les images  $F_t$  et  $F_{t+1}$  sera plus petite que toutes les autres distances entre les objets. Par conséquent, nous avons choisi d'associer des centroïdes à des distances minimales entre les images, dans le but de construire notre objet tracker afin de pouvoir calculer le centroïde du nouveau objet mais aussi l'associer un identifiant.



Figure 4: **calcul de la distance euclidienne**

Cette image est le résultat du calcul de la distance euclidienne ce qui nous a permis de calculer le centre de gravité des nouveaux objets sans perdre les autres objets déjà suivis, et on voit que le seuil de confiance nous a pas permis de détecter toutes les personnes dans la frame ce qui est dit au problème d'occlusion que tous les algorithmes de suivie présentent.



Figure 5: **Problème d'occlusion pour le suivi des objets**



Nous observons dans ce cas que le programme identifie bien les personnes mais la détection est moins bonne pour certaines personnes car le seuil de confiance utilisée permet d'effectuer une bonne détection au cas où il n'y a pas d'occlusion mais à la présence d'une occlusion il performe moins bien (cas de l'image 3 en bas où le suivi est presque parfait).

## 5.2 Comptage des personnes

Pour effectuer le comptage des personnes nous avons défini une ligne d'intersection qui sépare notre scène en deux parties; **Going down** pour les personnes sortants et **Going up** pour les personnes entrants. On peut remarquer cela à travers l'image de droite de la figure ci-dessous. En se basant sur le suivi de mouvement de la personne nous attribuons un ID sur chaque objet (personne) détecté. Nous avons défini aussi un compteur qui va compter en temps réel le nombre de personnes entrant et sortant.

### 5.2.1 Expérimentation



Figure 6: Début du comptage

Sur la scène ci-dessus nous remarquons deux personnes sortants et une personne entrant. A cet instant sur notre compteur on a:

- **Going up:** 3 personnes
- **Going down:** 5 personnes

Remarquons le changement au niveau de notre compteur sur la scène ci-dessous après que ces personnes aient traversé la ligne d'intersection.



Figure 7: Fin du comptage

Après la traversée de la ligne d'intersection, nous remarquons une incrémentation sur les valeurs initiales au niveau des compteurs. on a:

- **Going up:** 4 personnes
- **Going down:** 7 personnes

Nous pouvons donc conclure que notre compteur est capable de fonctionner en temps réel puisque à chaque fois qu'une personne dépasse la ligne d'intersection ,le compteur se met automatiquement à jour.

## 6 Conclusion

Au terme de ce projet , notre travail a consiste a mettre en place un programme qui consistera dans un premier temps détecter des personnes dans une vidéo, ensuite suivre le mouvement de ces personnes ,afin compter le nombre de personne détecter. Au niveau de la phase de détection nous avons choisit la méthode d'extraction de l'arrière plan (background soustraction) pour supprimer tous les objets non nécessaire autour de l'objet a détecter; cela permettra de bien suivre l'objet en mouvement. Pour le suivi de mouvement nous avons créer des tracker pour chaque objet détecter pour suivre l'objet pendant son mouvement,mais aussi être en mesure de pouvoir gérer lorsqu'un objet a été perdu, disparu ou a quitté le champ de vision .

Nous notons que la détection des objets est relativement efficace mais est perturbée par la proximité des couleurs des objets avec l'arrière plan ou par les occlusions. Le suivi quant à lui n'est vraiment correct que pour des objets uniques ou ceux qui sont visibles dans la scène.. Ce Tp aura été en somme très instructif et nous aura permis d'acquérir de multiples connaissances en matière de détection et de suivi de mouvements dans les vidéos.

## References

- [1] <https://pdfs.semanticscholar.org/25a6/c5dff9a7019475daa81cd5a7f1f2dcdb5cf1.pdf>
- [2] <https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/>  
 lien pour le modèle pre-entraîné:  
[Pre-trained model](#)