

# Vision par Ordinateur: TP2 : Compter des passagers

CIBAMBO Masugentwali Steven & DEGUE Akouvi Mireille  
*Enseignant : NGUYEN Thi Oanh (Ph.D)*

Décember 2019

## 1 Introduction

La vision par ordinateur est une branche de l'intelligence artificielle dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition. Ainsi plusieurs systèmes trouvent leur place dans cette branche entre autre la classification d'images, la reconnaissance faciale, la détection et le suivi de mouvement, etc.

Dans ce rapport nous présentons les résultats obtenus sur la technique de détection et suivi d'objets en mouvement dans une scene que nous avons réaliser avec python 3.7, opencv 3.4 et le filtre de Kalman pour le suivi d'objets en mouvement. La solution proposée est applicable sur un flux vidéo du webcam et aussi sur une vidéo pré-enregistrée.

### 1.1 Exécution

Code source : [sur github](#)

`python motion_detection.py -video chemin_du_fichier`

pour voir le résultat du filtre de Kalman `python measured.py`

ou

`python motion_detection.py` si vous voulez utiliser le webcam

## 2 Détection de mouvement

### 2.1 Soustraction du Background

La soustraction du background est critiale dans la plus part de applications de Vision par ordinateur, on peut l'utiliser pour savoir le nombre de véhicules qui passent dans la route, ou les personnes et aussi on peut l'utiliser pour la détection de mouvement et autre applications. L'arrière-plan de notre flux vidéo est en grande partie statique et immuable sur les images consécutives d'une vidéo. Par conséquent, si nous pouvons modéliser l'arrière-plan, nous le surveillons pour des changements substantiels. S'il y a un changement substantiel, nous pouvons le détecter; ce changement correspond normalement au mouvement sur la scene. Maintenant, évidemment, dans le monde réel, cette hypothèse peut facilement échouer. En raison de l'ombrage, des réflexions, des conditions d'éclairage et de tout autre changement possible dans l'environnement, notre arrière-plan peut être très différent dans différentes images d'une vidéo. Et si l'arrière-plan semble différent, cela peut perturber l'algorithme. C'est pourquoi les systèmes de détection par soustraction de background les plus performants utilisent des caméras montées fixes et dans des conditions d'éclairage contrôlées. Dans notre cas, si la vidéo est donnée en parametre pour la détection notre système utilise la première image de la vidéo comme background dans le cas contraire, si le flux vient du webcam le système utilise la première vue comme background. Dans le deux cas si le background change pour une raison ou une autre cela pertube l'algorithme comme dit précédemment.



Figure 1 (a)

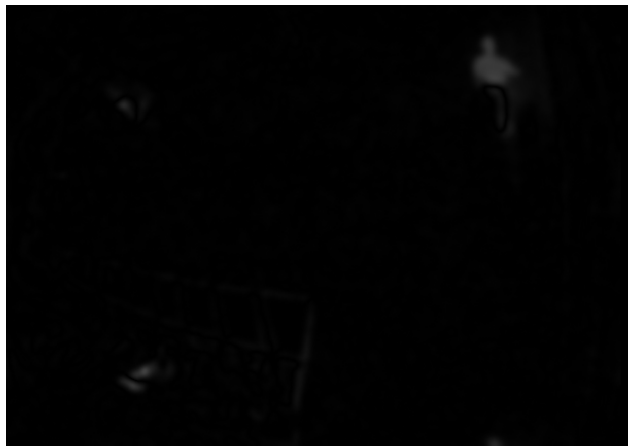


Figure 1 (b)

La première figure montre bien le résultat obtenu en faisant la soustraction du background et une autre image de la vidéo figure 1 (a). Mais le résultat n'est pas très clair, nous proposons de faire un seuillage enfin de garder que les parties en mouvement.

## 2.2 Seuillage, érosion et dilatation

Au moment où notre background est static nous sommes prêt à détecter tout objet en mouvement sur la scene. Ainsi nous effectuons une érosion et une dilatation pour elliminer tout défaut qui resiste encore.



Figure 2 (a)



Figure 2 (b)



Figure 2 (c)



Figure 2 (d)

Sur la figure 2 nous remarquons à gauche (a et c) l'image issue de la soustraction du background et une autre, à droite faisons le seuillage enfin de bien observer le changement.

### 2.3 Objet en mouvement

La détection de mouvement basée sur la soustraction du background consiste à garder la partie statique soit le background en noir et les parties sur lesquelles les mouvements sont détectés en blanc. Dans notre cas nous avons fixé la taille minimum de l'objet en mouvement enfin de l'encadrer en vert comme le montre la figure 3. Ce qui est plus intéressant en est que, même si l'objet en mouvement n'est pas encadré par ce que sa taille est inférieure à ce que nous avons fixé cela est toujours détecter si on fait la soustraction du background, figure 3 (b et d)



Figure 3 (a)

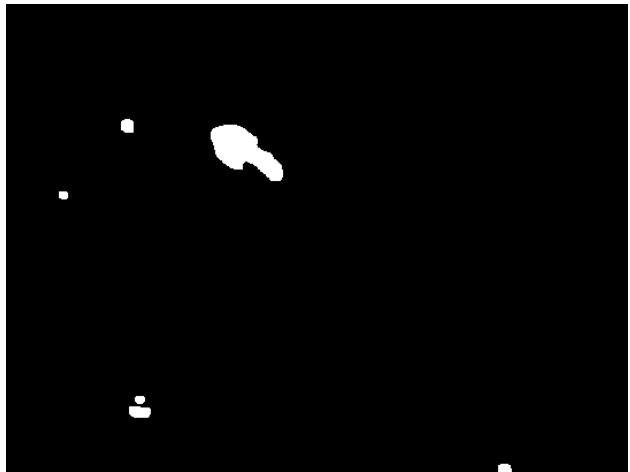


Figure 3 (b)



Figure 3 (c)

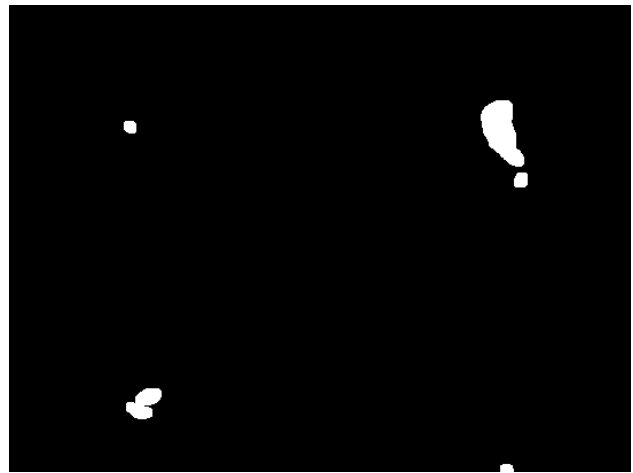


Figure 3 (d)

Dans le cas général de vidéo pré-enregistrée la première image est noir, cela fait à ce que, s'il y a un objet en mouvement dans la scene on a présence de la couleur blanche quel que soit sa taille.

### 3 Suivi de mouvement

Dans cette deuxième partie, nous montrons le résultat obtenu pour le suivi d'objets en mouvement. Pour cela, nous avons utilisé le Filtre de Kalman pour la prédiction et la correction de position de l'objet en mouvement. Les différentes positions de l'objet en mouvement mesurées sont stockées dans un tableau et pour tout index du tableau où aucun mouvement n'est détecté sa valeur vaut -1, c'est le même cas pour un objet qui passe derrière ou en dessous d'un objet statique; par exemple une balle qui roule en dessous d'une table. Cela veut dire pour les positions non mesurées le Filtre de Kalman peut les estimer avec précision.



Figure 4 (a)

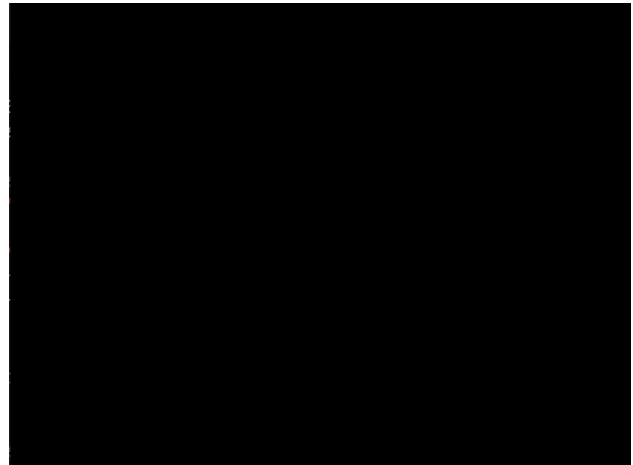


Figure 4 (b)



Figure 4 (c)

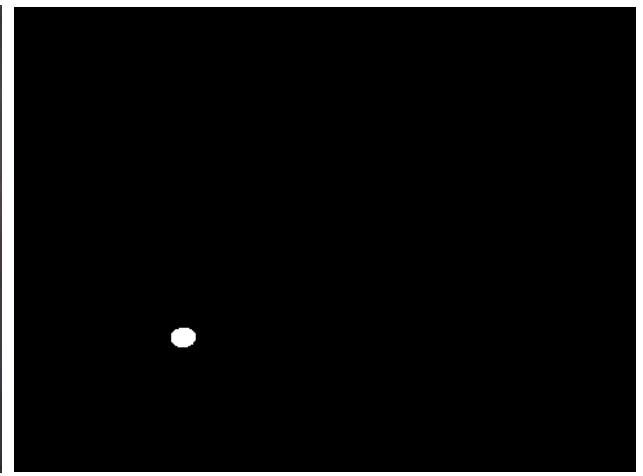


Figure 4 (d)

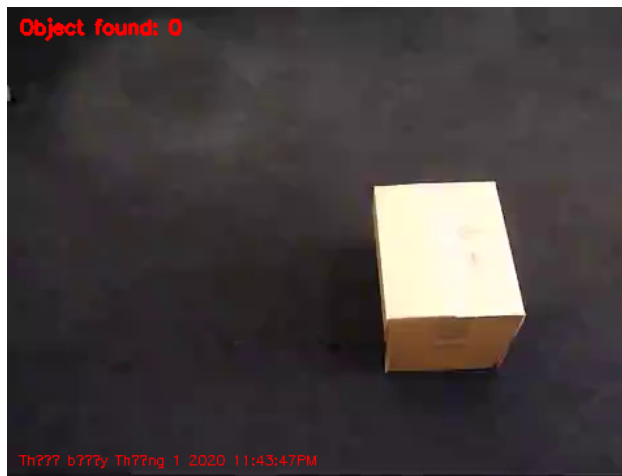


Figure 4 (e)



Figure 4 (f)



Figure 4 (g)

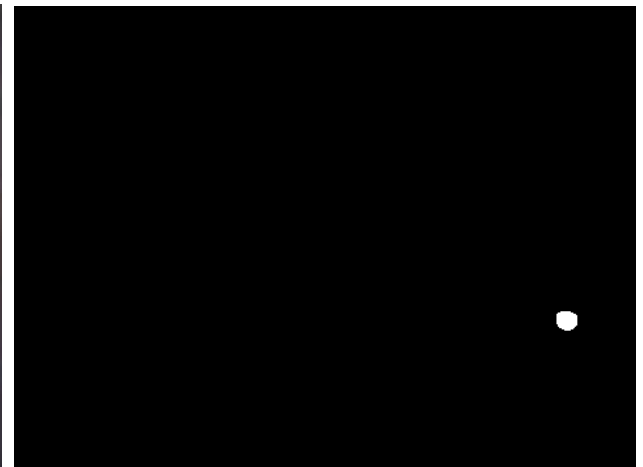


Figure 4 (h)

Les images de la figure 4, sont capturées dans une vidéo montrant une balle qui roule et à un certain moment la balle passe en dessous d'une table. Nous considérons cette vidéo juste pour illustrer les fonctionnement

du Filtre Kalman; mais plus loin nous prenons d'autres vidéo pour l'experimentation enfin de voir les limites de notre système. Sur la figure 4 (a) voyons que, aucun mouvement n'est détecté sur la scene comment le montre la figure 4 (b) fond tout noir. Sur la figure 4 (c) un objet en mouvement est détecté et une mesure de sa position est prise au fur et mesure que l'objet se deplace. Mais à un certain moment la caméra perd de vue l'objet figure 4 (e) et réapparaît plus tard figure 4 (g). Non seulement les positions mesurée peuvent contenir certaines erreurs mais il faut aussi estimer les positions quand l'objet disparaît sur la scene à un certain moment. Pour y arriver, le filtre de Kalman a besoin de coordonnées de la position courante de l'objet soit ( $p_x$  et  $p_y$ ) mais aussi de la vitesse courante de l'objet soit ( $v_x$  et  $v_y$ )

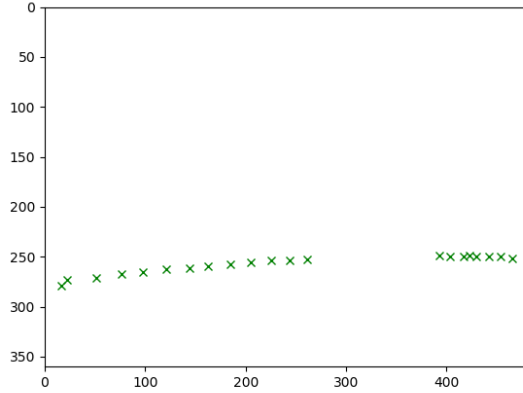


Figure 4 (i)

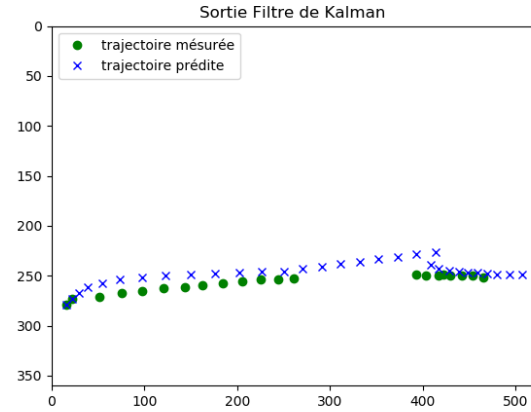


Figure 4 (j)

Sur la figure 4 (i) pour les index du tableau où aucun objet en mouvement n'est détecté ça vaut -1 et aucun marqueur n'y est posé, cela est remarqué au début de la vidéo et au moment où la balle passe en dessous de la table. En appliquant le filtre de Kalman sur les positions mesurées, nous pouvons prédire ces différentes positions comme le montre la figure 4 (j). Sur la figure suivante nous montrons la prédiction de Kalman sur un vidéo avec beaucoup d'objets en mouvement.

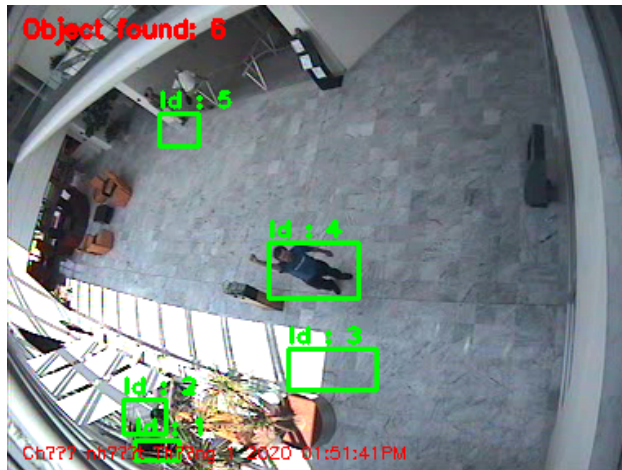


Figure 5 (a)



Figure 5 (b)



Figure 5 (c)



Figure 5 (d)

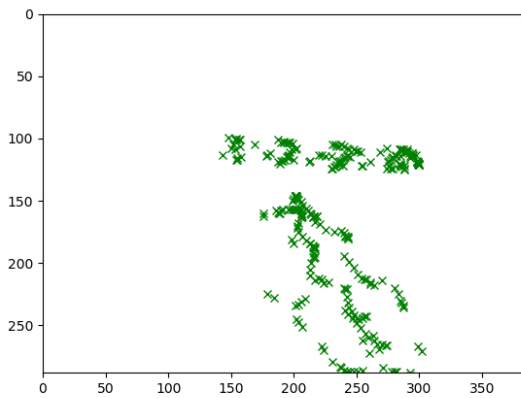


Figure 5 (e)

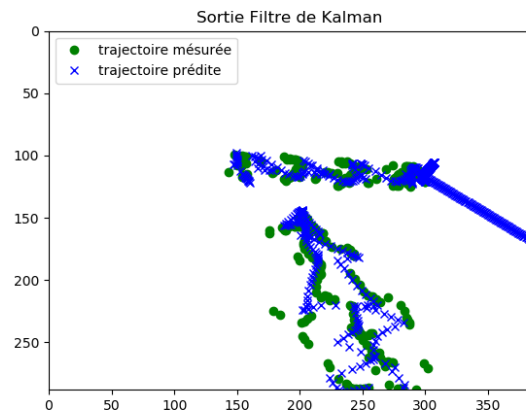


Figure 5 (f)

Les images de la figure 5 sont issues d'une vidéo avec plusieurs objets en mouvement ce pour cela que sur figure 5 (e) nous voyons beaucoup de marqueurs.

**Point à souligné;** sur la figure 5 (c) la personne avec id : 1, est venu et resté debout pendant longtemps à une position bien fixe; mais le filtre de Kalman continu à prédire la trajectoire suivant la vitesse courante et la position de l'objet en mouvement comment le montre la figure 5 (f); cela par ce que la vidéo n'a pris fin mais l'objet en mouvement s'est arrêtée.

## 4 Comptage d'Objets en mouvement

Pour parvenir à compter les objets qui sont en mouvement sur la scene, nous nous sommes servis de la technique de detection de contours dans chaque frame et pour chaque contour détecté nous l'identifions par un marqueur unique.



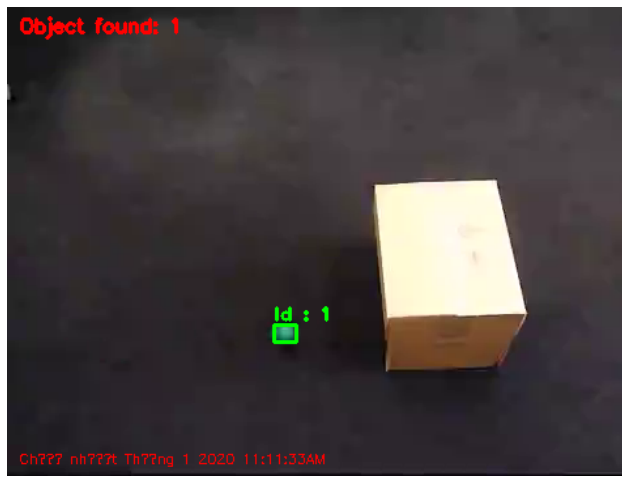


Figure 5 (a)

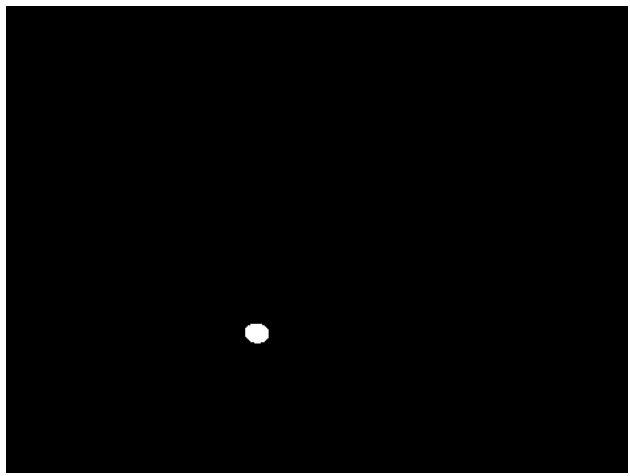


Figure 5 (b)

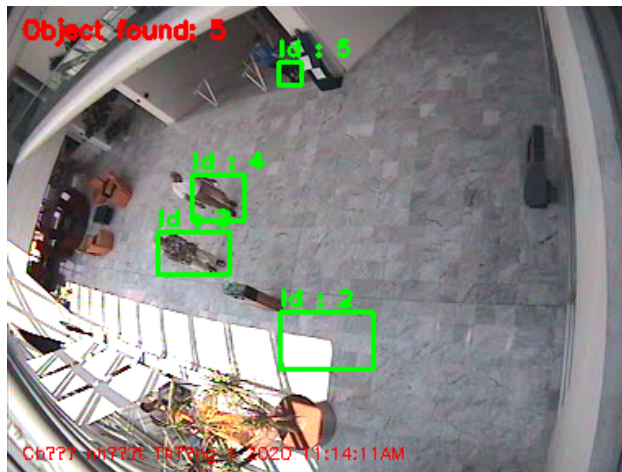


Figure 5 (c)



Figure 5 (d)

Dans chaque frame de la vidéo le nombre de contours peut varier suivant le nombre d'objet en mouvement du fait que nous utilisons la détection par soustraction de background.

## 5 Conclusion

La détection et suivi d'objet en mouvement dans une scene de vidéo est un concept très utilise et rependu actuellement; mais il devien encore plus interessant quand on prend tout le parametre en compte entre autre l'environnement, la nature d'objets, type de caméra, etc. Dans ce travail nous nous sommes basé plus sur;

- Détection d'objets en mouvement: plusieurs technique existe actuellement pour la détection d'objet en mouvement avec avantage et inconvénient chacune mais nous avons utilisé la soustraction du background et pour bien différencier les zones en mouvement et statique nous avons fait seuillage (binarization), érosion et la dilatation.
- Suivi d'objet en mouvement: pour parvenir à suivre un objet en mouvement ce n'est pas évident car souvent la position mesurée peut contenir aussi bien des erreurs; ainsi nous avons utilisé le filtre de Kalman implémenter dans la librairie python pykalman enfin de prédire et corriger ces erreurs.
- Comptage d'objet en mouvement sur la scene: vue la divergence de videos utilisées pendant l'experimentation pour faire simple nous avons utiliser la detection de contours, et comme nous utilisons la soustraction du background dans chaque frame cela marche bien et enfin chaque objet en mouvement est identifier un numéro unique dans la scene.



## Références

- Computer Vision. *“Computer Vision with Python.”* Computer Vision. December 2019.
- Garima Nishad. *“Kalman Filters: A step by step implementation guide in python”* December 2019.
- Adrian Jackson. *“Multi-object tracking with dlib”* December 2019.