

## REPRISE MODULE

---

VISION PAR ORDINATEUR - Promo 24 - Juillet 2021

### TP3 – Segmentation des objets en se basant sur le flux optique

**Préparé par:** Jean-Bertrand Fritzner Leon

**Supervision :** Dr. Nguyen Thi Oanh

---

#### Introduction

Avec la généralisation de l'utilisation d'images numériques, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme machine, l'analyse de séquences sportives... En effet, les zones de mouvement d'une séquence d'images correspondent souvent à des événements sur lesquels un système de vision doit se focaliser. L'analyse du mouvement est un vaste sujet qui englobe un certain nombre de problématiques. On peut notamment citer :

- la détection du mouvement, qui consiste à étiqueter chaque pixel d'une image suivant s'il correspond ou non à une région en mouvement dans la scène,
- la détection des objets en mouvement, c'est-à-dire la détection d'un ensemble de régions d'intérêt en mouvement dans la scène tridimensionnelle observée
- la segmentation basée mouvement de la scène, pour laquelle chaque région de l'image ayant un mouvement distinct des autres est détectée et segmentée,
- l'estimation du mouvement, qui consiste à estimer, à partir d'une séquence d'images, le mouvement apparent des objets composant une scène tridimensionnelle,
- le suivi de primitives ou de régions, dont le but est de déterminer la position de chaque

primitive ou région dans l'image à chaque instant,  
la reconnaissance et la modélisation d'activités ou de gestes.

Les trois premières problématiques (détection du mouvement, détection des objets en mouvement et segmentation basée mouvement), qui sont au cœur des travaux présentés, sont en général une première étape pour des outils automatiques de vision par ordinateur. Ces outils peuvent avoir pour vocation, soit uniquement de détecter, soit de détecter et reconnaître, soit de détecter et suivre des objets d'où le calcul du flot optique qui est une étape de bas niveau en traitement d'images, permettant d'estimer le déplacement des objets d'une scène. Toutefois, les estimations denses et précises du flot optique sont habituellement coûteuses en temps de calcul. Notre travail s'attachera donc à faire une estimation du flot optique dans une séquence d'images et segmenter des objets en mouvement.

## Objectif

- Pour chaque frame d'une vidéo, nous allons faire une estimation du flot optique de ce frame en se basant sur le frame courant et le frame précédent.
- Utiliser ces valeurs dans la deuxième partie de notre travail.
- Visualiser la norme en créant une image dont la valeur du pixel est la norme de son flot optique.
- Segmenter des objets en mouvement en appliquant une méthode de segmentation (seuillage, k-means, ...)
- Utilisez plusieurs seuils pour segmenter des objets à différentes vitesses.

## 1.- Première partie : Estimation du flot optique dans une séquence vidéo

Dans cette partie, on s'attachera à décrire les approches de calcul du flot optique **Mouvement image** : projection du mouvement 3D réel de la scène dans le plan 2D de l'image.

**Flot optique** : champ des vitesses mesuré à partir des variations de la luminance.

La plupart des méthodes d'estimation du flot optique reposent sur une hypothèse fondamentale : l'intensité lumineuse (ou une autre variable photométrique) se conserve entre deux images successives.

Cela s'écrit sous la forme générale :  $I(x+w, t+1) - I(x, t) = 0$

→ Nous allons tout d'abord utiliser la fonction `cv.calcOpticalFlowPyrLK()` pour suivre les

points caractéristiques de la vidéo.

Nous passons l'image précédente, les points précédents et l'image suivante. Il renvoie les points suivants avec des numéros d'état qui ont une valeur de 1 si le point suivant est trouvé, sinon zéro. Nous passons itérativement ces points suivants comme points précédents à l'étape suivante.

En déduire l'implémentation ci-dessous :

```
#flow process
next, status, error = cv.calcOpticalFlowPyrLK(prev_gray, gray, prev, None, **lk_params)
good_old = prev[status == 1]#position variables
good_new = next[status == 1]#position variables
#marking flow tracks
for i, (new, old) in enumerate(zip(good_new, good_old)):
    a, b = new.ravel()
    c, d = old.ravel()
    mask = cv.line(mask, (a, b), (c, d), color, 1)
    frame = cv.circle(frame, (a, b), 2, color, -1)
output = cv.add(frame, mask)
prev_gray = gray.copy()
prev = good_new.reshape(-1, 2)
```



Image original



Point de carateristique

→ Nous allons créer un champ de flux optique dense en utilisant la méthode **cv.calcOpticalFlowFarneback()**, fonction déjà implémenté dans la bibliothèque OpenCV, de chaque sequence ou frame d'image dans une vidéo

Marche à suivre :

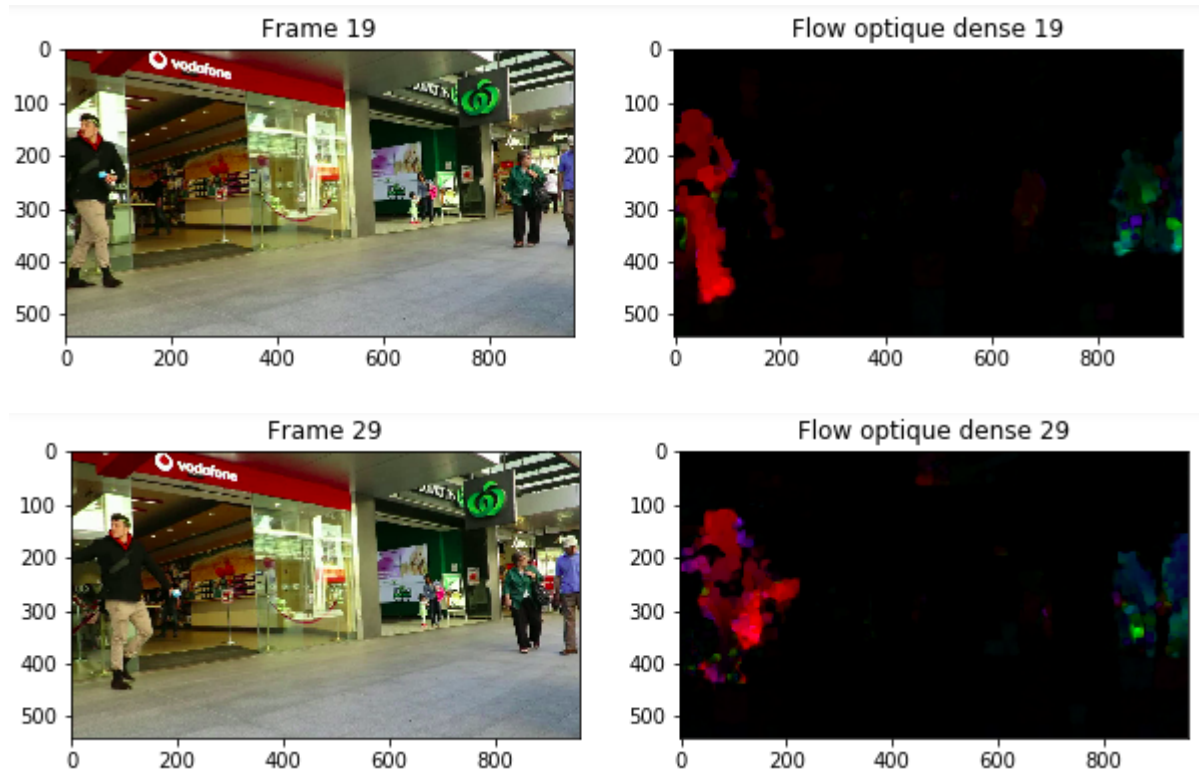
- Lecture de la vidéo
- Ensuite récupérer le frame précédent en niveau de gris
- Finalement on visualise les frames courant et de son flow optique

```
#dense optical flow
flow = cv.calcOpticalFlowFarneback(prev_gray, gray, None, 0.5, 3, 15, 3, 5, 1.2, 0)
magnitude, angle = cv.cartToPolar(flow[..., 0], flow[..., 1])
mask[..., 0] = angle * 180 / np.pi / 2

# Sets image value according to the optical flow magnitude
mask[..., 2] = cv.normalize(magnitude, None, 0, 255, cv.NORM_MINMAX)

# Converts HSV to RGB (BGR) color representation
```

A chaque pixel du frame, nous avons un vecteur de vitesse ( flot optique) dont l'angle (direction) et la norme peuvent être déterminés.



On obtient comme résultat des frames courant et son flow optique.

Etant donné que notre vidéo dure environs plus d'une vingtaine de secondes on aura donc une quantité de frames et de flot optique car dans une seconde on peut capturer entre 25 et 30 scènes. Nous allons vous présenter quelques captures .

## 2.- Deuxième partie : Segmentation des objets en mouvement

La segmentation des objets en mouvement dans une séquence d'images peut être formulée comme un problème de soustraction d'arrière-plan - la séparation des objets de l'arrière-plan dans chaque cadre d'image. La scène de fond est apprise et modélisée. Un processus pixel par pixel est utilisé pour classer chaque pixel en tant qu'objet ou arrière-plan en fonction de sa similitude avec le modèle d'arrière-plan.

En outre la définition formelle de la segmentation est la subdivision (partitionnement) de la vidéo en plusieurs régions  $R_i, i \in \{1, \dots, K\}$ , telles que :

La qualité de la segmentation dépend de la fiabilité de la caractéristique que l'on considère pour

- $\bigcup_{i=1}^K R_i = \Omega$ , où  $\Omega$  est le domaine de la vidéo.
- $R_i \cap R_j = \emptyset$ , si  $i \neq j$ .

l'appartenance à une région donnée. Dans le cas de l'estimation du mouvement, la segmentation

souffrira des même lacunes que celle-là, notamment :

- Problème d'ouverture
- Changement d'illumination
- Occlusion
- Grands mouvements

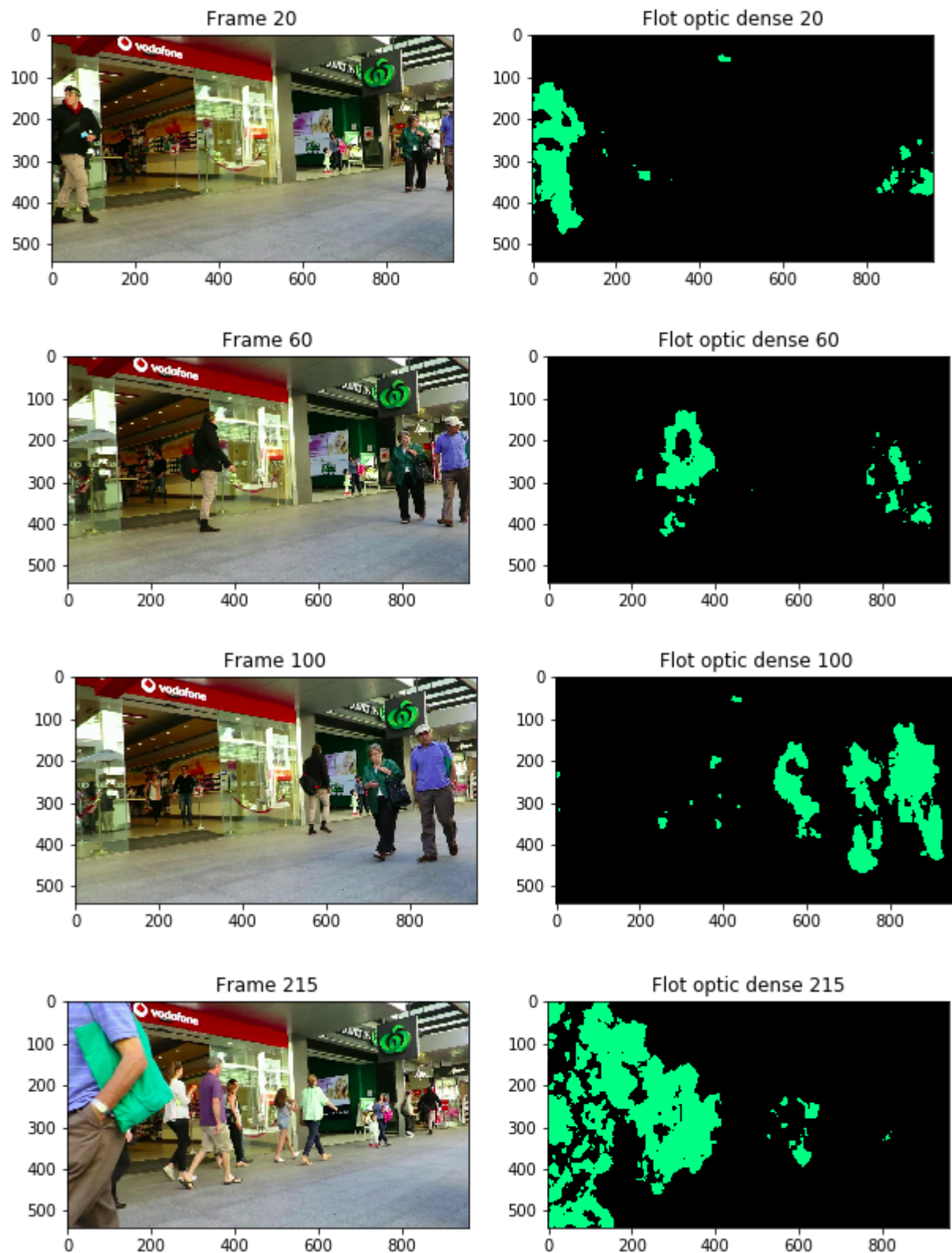
La segmentation s'effectue dans le domaine spatio-temporel, qui définit une séquence vidéo.

Donc dans cette étape, nous allons segmenter les objets qui se déplacent le plus vite en appliquant le seuillage sur les normes du flot optique. Ensuite, nous allons utiliser plusieurs seuils pour segmenter des objets à différentes vitesses.



## Expérimentation et analyse des resultats

### Résultat avec 1 seuil et interprétation:



La durée de la vidéo est de 30 secondes, avec une vitesse réduite pour le traitement de chaque

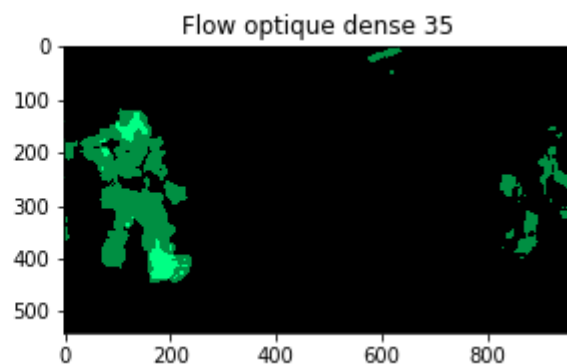
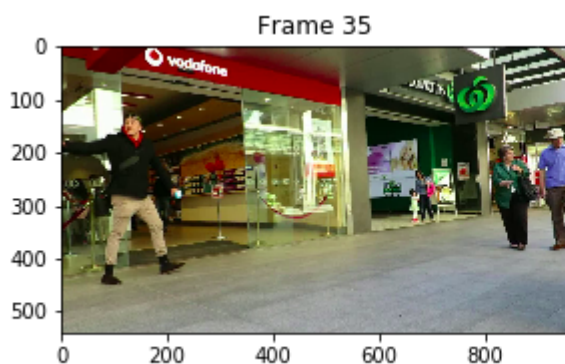
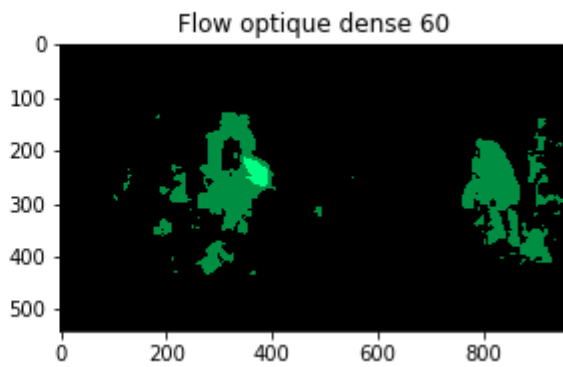
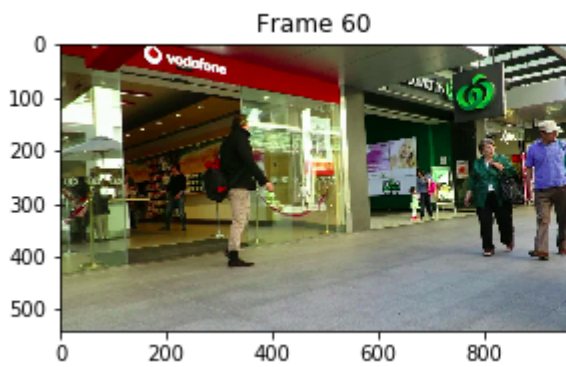
frame, cela a pu prendre un peu plus de temps pour faire le traitement et parmi les résultats obtenus nous avons tiré quelques-uns, les images de droites montre la segmentation des piétons en mouvement au bord d'un centre commercial. Alors que les petits coûts de bord entraînent des segments avec seulement quelques pixels, les coûts de bord élevés entraînent de petites régions. calcul du flow à partir du frame =  $\{0.5, 3, 10, 3.5, 0.5, 0\}$ . Pour raffiner le résultat de segment, et palier au problème de trou ; lors de la détection du mouvement, nous avons supprimé les petites régions facultatives et convertis en jouant sur l'effet des paramètres pour assurer que les couleurs sont correctement transmises pour afficher les images à l'aide de matplotlib.

### Résultat avec 2, 3 seuil et interprétation:

Seuil respectivement :

s1=30

s2=180

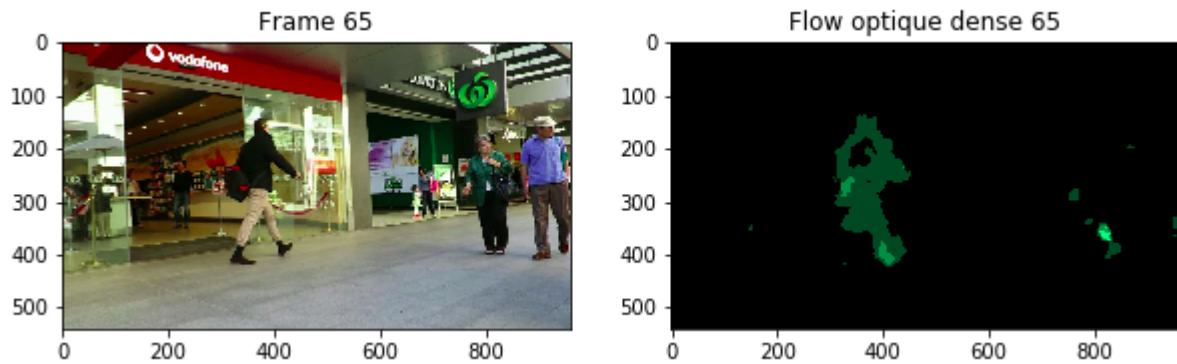


Seuil respectivement :

s1=64

s2=128

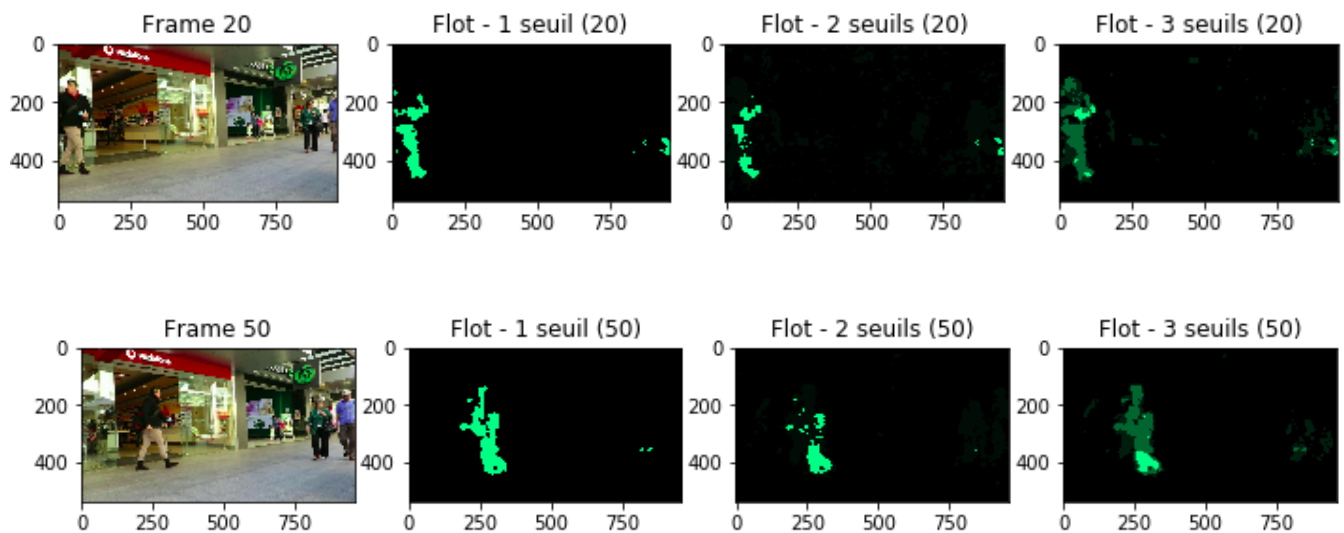
s3=192



En visualisant les figures ci-dessus on détecte des bons et mauvais résultats compte tenu du seuil en question.

### Comparaison des correspondances de suivie

A travers ces images, nous allons vous montrer un exemple suite aux différents types de seuillage accorde pour la détection de chaque frame., on estime qu'on détecte le mieux avec un seuil, donc la variation dépend de vous (choix aleatoire).



### Outils et base de vidéo utilisé

Nous avons utilisé une vidéo du site [motchallenge.com](http://motchallenge.com) qui permet de suivre le mouvement des objets par la suite nous avons utilisé un ensemble d'outils pour l'implémentation en python, jupyter notebook avec des bibliothèques comme Opencv, matplotlib, numpy, tensor flow.



## Discussion et Conclusion

Dans tous les exemples précédents, nous avons déterminé les seuils de manière à minimiser la variance, c'est-à-dire que nous avons appliqué à nos mesures la méthode de binarisation proposée pour des images en niveaux de gris ce qui implique que les seuils ont une influence conjointe sur les résultats, comme les séquences vidéo ont des dimensions différentes, et les objets recherchés sont hétérogènes du point de vue de leur taille et de leur vitesse de déplacement, nous avons choisi la longueur des séquences élémentaires et la taille des blocs spatio-temporels en fonction de ces particularités.

Notre travail consistait à faire une estimation du flot optique dans une séquence d'images et segmenter des objets en mouvement, ce dont nous avons fait afficher quelques résultats.

## Référence

<https://hal.archives-ouvertes.fr/hal-01653720v2/document>

<https://www.cvl.isy.liu.se/en/research/objrec/moving-object-segmentation/index.html>

[https://www.irisa.fr/vista/Papers/2007\\_cvpr\\_bugeau.pdf](https://www.irisa.fr/vista/Papers/2007_cvpr_bugeau.pdf)

[https://www.researchgate.net/publication/37987320\\_Moving\\_Object\\_Segmentation\\_Using\\_Optical\\_Flow\\_and\\_Depth\\_Information](https://www.researchgate.net/publication/37987320_Moving_Object_Segmentation_Using_Optical_Flow_and_Depth_Information)

[https://vision.in.tum.de/\\_media/spezial/bib/wedel\\_et\\_al\\_emmcvpr09.pdf](https://vision.in.tum.de/_media/spezial/bib/wedel_et_al_emmcvpr09.pdf)

code projet : <https://github.com/birthou/Optical-flow-Tp>