INSTITUT FRANCOPHONE INTERNATIONAL

MODULE VISION PAR ORDINATEUR - M2 - Promo 24 - Mars 2021

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

Préparé par: Jean-Bertrand Fritzner Leon & Michelet Juste (Groupe 4)

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 hommemachine, 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 si 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 composants 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 précédant.
- 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.

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 : $I(x+\omega, t+1) - I(x, t) = 0$

- 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 deduire l'implementation 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, 1, 2)
```

Image original

Point de carateristique

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

Marche a suivre :

- Lecture de la vidéo
- Ensuite recupérer le frame precedant en niveau de gris
- Finalement on visualise les frames courrant 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
rgb = cv.cvtColor(mask, cv.COLOR_HSV2BGR)

#window shows output conversion flow
cv.imshow("dense optical flow", rgb)
prev_gray = gray
#window distruction function pess 'q'
if cv.waitKey(1) & 0xFF == ord('q'):
    break
```

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 resultat des frames courrant et son flow optique.
Etant donné que notre vidéo dure environs plus d'une vingtaine de secondes on aura donc une quantite de frames et de flow optique car dans une seconde on peut capturer entre 25 et 30 scènes. Nous allons vous présenter quelque captures .

Deuxieme partie : Segmentation des objets en mouvement

La definition formelle de la segmentation est la subdivision (partitionnement) de la video en plusieurs régions $R_i, i \in \{1, \ldots, K\}$, telles que :

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

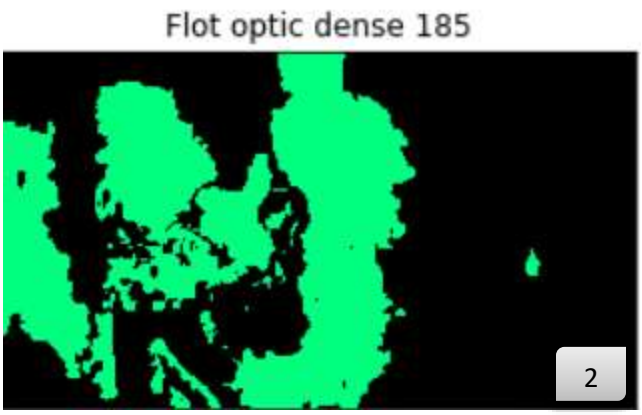
La qualite de la segmentation dépend de la fiabilité de la caractéristique que l'on considere pour l'appartenance à une région donnée. Dans le cas de l'estimation du mouvement, la segmentation souffrira des même lacunes que celle-la, notamment :

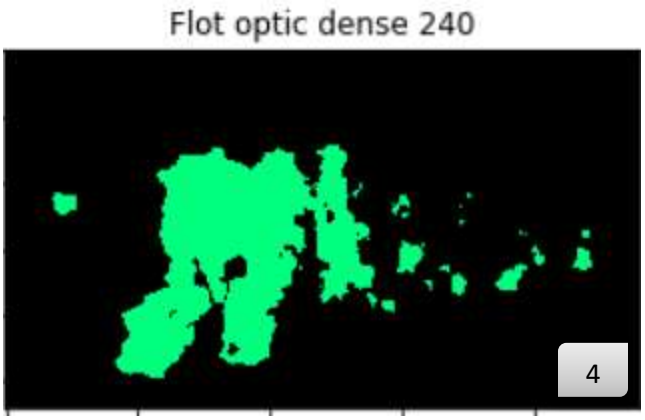
- Problème d'ouverture
- Changement d'illumination
- Occlusion
- Grands mouvements

La segmentation s'effectue dans le domaine spatio-temporel, qui definie une séquence vidéo.
Donc dans cette etape, nous allons segmenter les objets qui se deplace 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.

Experimentation et analyse des resultats

Resultat avec 1 seuil :

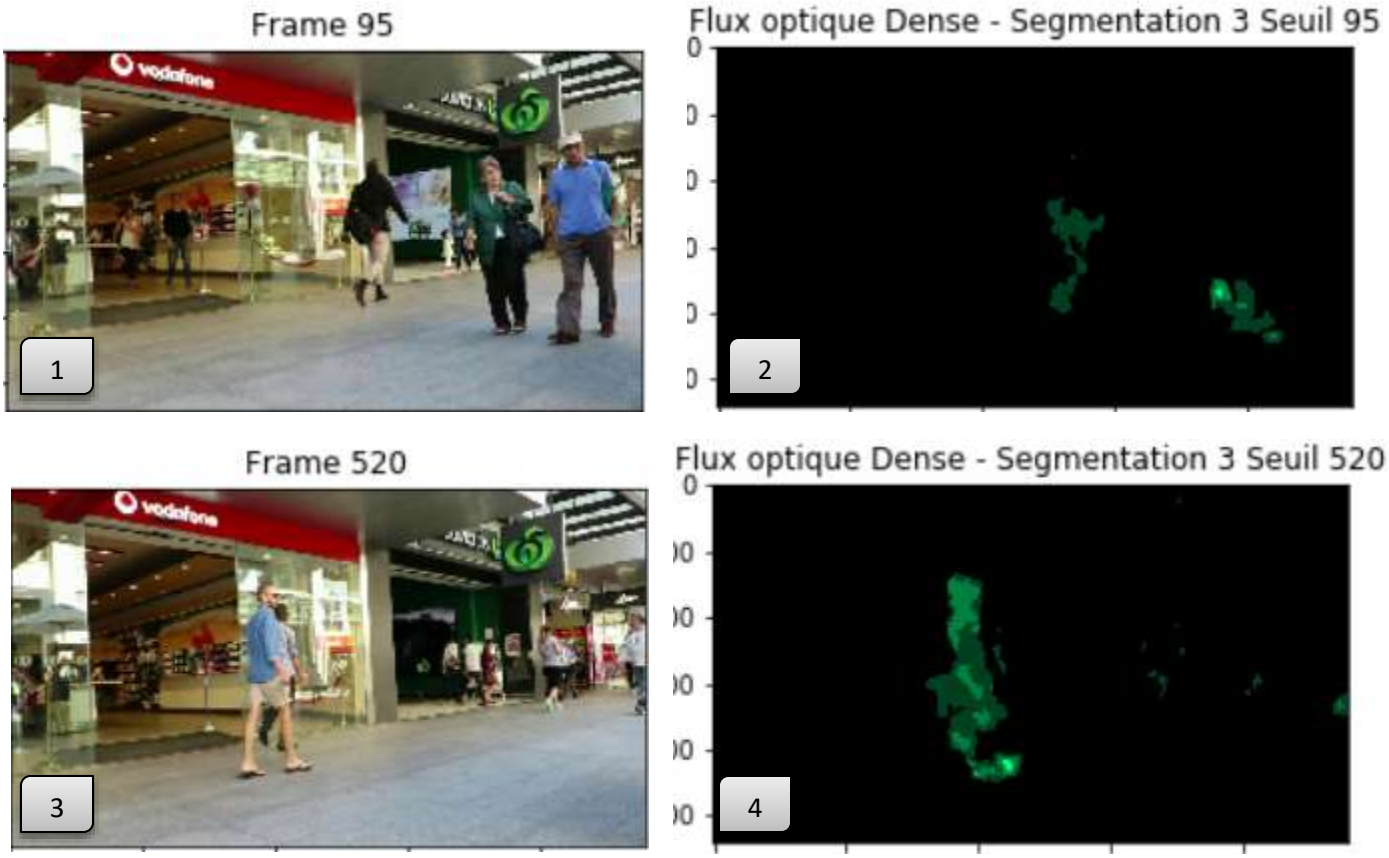




Explication 1:

La durée de la video est de 30 secondes, avec une vitesse reduite pour le traitement de chaque frame, cela a pu prendre un peu plus de temps pour faire le traitement et parmi les resultats obtenus nous avons tiré quelques-un, les images de droites montre la segmentation des pietons en mouvement au bord d'un centre commercial. Alors que les petits coûts de bord entraînent des segments avec seulement quelques pixels (images 2 et 4), 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 resultat de segment, et palier au probleme de trou ;lors de la detection du mouvement, nous avons supprimer les petites regions 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 3 seuil :



Explication 2:

En visualisant l'image 2 et 4 de notre figure nous avons vu que le resultat de la detection était mauvais pour le frame 95 et 520, pourtant correct dans d'autre (juste pour montré une mauvaise detection), surement ca aurait du a la luninosité, les trous a l'interieur des objets homogènes à déplacement lent, en raison du probleme d'ouverture.

Outils et base de video utilisé

Nous avons utilisé une video du site Motchallenge.com qui permet de suivre le mouvement des objets par la suite nous avons utilisé un ensemble d'outils pour l'implementation en python, jupyter notebook avec des bibliotheques comme OpenCV, Matplotlib, Numpy ect.

Discussion et Conclusion

Dans tous les exemples precedents, nous avons determine les seuils de maniere a minimiser la variance, c'est-`a-dire que nous avons appliqu a nos mesures la methode de binarisation proposee pour des images en niveaux de gris ce qui implique que les seuils ont une influence conjointe sur les resultats, comme les sequences video ont des dimensions diff´erentes, et les objets recherches sont heterogenes du point de vue de leur taille et de leur vitesse de deplacement, nous avons choisi la longueur des sequences elementaires et la taille des blocs spatio-temporels en fonction de ces particularites.

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.

Lien vers le code du projet :
https://github.com/birthou/Optical_flow-Segmentation