

TP2

La soustraction d'arrière-plan est l'une des principales tâches de traitement d'image. Il est utilisé dans diverses applications de traitement d'images telles que la segmentation d'images, la détection d'objets, etc. OpenCV nous fournit 3 types d'algorithmes de soustraction d'arrière-plan :

- BackgroundSubtractorMOG
- BackgroundSubtractorMOG2
- BackgroundSubtractorGMG

La bibliothèque OpenCV prend en charge la version MOG2, cependant, pour les deux autres versions MOG et GMG, vous devez installer les dépendances **contrib** pour que cela fonctionne car cela ne fait pas partie de la construction standard.

```
pip install opencv-contrib-python --user
```

1. Activité n°1 : Modélisation de l'arrière-plan par MOG

Pour utiliser **BackgroundSubtractorMOG**, nous pouvons utiliser :

```
cv2.bgsegm.createBackgroundSubtractorMOG()
```

Ensuite, nous pouvons l'appliquer en utilisant la méthode **apply()** sur chaque frame de la vidéo.

```
import cv2
cap = cv2.VideoCapture(0)
# initializing subtractor
fgbg = cv2.bgsegm.createBackgroundSubtractorMOG()
if (cap.isOpened()== False):
    print("erreur d'ouverture de vidéo")
while(cap.isOpened()):
    ret, frame = cap.read()
    # applying on each frame
    fgmask = fgbg.apply(frame)
    cv2.imshow('frame', frame)
    cv2.imshow('MOG', fgmask)
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

- 1.1. Evaluer les paramètres optionnels de la méthode **createBackgroundSubtractorMOG()** (int history,int nmixtures, double backgroundRatio, double noiseSigma)

2. Activité 2 : Modélisation de l'arrière-plan par MOG2 (Mixture of Gaussians Version 2 - OpenCV 3.x+)

BackgroundSubtractorMOG2 est une version améliorée de MOG, toujours basée sur le modèle de mélange gaussien, mais avec plusieurs améliorations. **BackgroundSubtractorMOG2** prend en charge la détection d'ombres et gère mieux des variations d'éclairage soudaines ou des mouvements complexes dans la scène. Pour appliquer **BackgroundSubtractorMOG2**, on utilise :

```
cv2.createBackgroundSubtractorMOG2()
```

- 2.1. Varier les trois paramètres **history**, **varThreshold** et **detectShadows** et déduire leurs impacts sur la localisation des objets mobiles

3. Activité 3 : Modélisation de l'arrière-plan par GMG (Geometric Mean and Gradient)

BackgroundSubtractorGMG utilise une approche combinant un modèle de fond probabiliste et des informations sur le gradient d'image pour détecter les objets en mouvement. Cet algorithme est plus sophistiqué et combine des techniques de traitement probabiliste et de filtrage basé sur les gradients. Pour implémenter **BackgroundSubtractorGMG**, on utilise :

```
cv2.bgsegm.createBackgroundSubtractorGMG()
```

```
import cv2
cap = cv2.VideoCapture(0)
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
# initializing subtractor
fgbg = cv2.bgsegm.createBackgroundSubtractorGMG()
if (cap.isOpened()== False):
    print("erreur d'ouverture de vidéo")
while(cap.isOpened()):
    ret, frame = cap.read()
    # applying on each frame
    fgmask = fgbg.apply(frame)
    fgmask = cv2.morphologyEx(fgmask, cv2.MORPH_CLOSE, kernel)
    cv2.imshow('frame', frame)
    cv2.imshow('GMG', fgmask)
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

- 3.1. Evaluer les paramètres optionnels de la méthode **createBackgroundSubtractorGMG ()** (int **initializationFrames = 120**, double **decisionThreshold = 0.8**)
- 3.2. Modifier l'élément structurant (forme et taille) et voir son influence sur le résultat obtenu.

4. Activité 4 : Modélisation de l'arrière-plan par KNN

Pour réaliser une segmentation avant-plan/arrière-plan en utilisant KNN (K-Nearest Neighbors), OpenCV fournit une implémentation pratique du **BackgroundSubtractorKNN**.

```
knn_subtractor = cv2.createBackgroundSubtractorKNN(history=100, dist2Threshold=100, detectShadows=True)
```

Ce modèle KNN permet de détecter les mouvements dans la vidéo et de séparer les objets en mouvement (avant-plan) de l'arrière-plan en temps réel.

Le KNN utilisé par OpenCV pour la soustraction de fond fonctionne en modélisant chaque pixel en fonction de ses valeurs antérieures et en utilisant les voisins les plus proches pour déterminer s'il appartient à l'avant-plan ou à l'arrière-plan.

La valeur de K utilisée par **createBackgroundSubtractorKNN** est fixée à **K = 2** par défaut dans la version actuelle d'OpenCV.

Cela signifie que l'algorithme regarde les 2 plus proches voisins dans l'historique des frames pour chaque pixel pour décider si un pixel appartient à l'arrière-plan ou au premier plan (objet en mouvement)

```
import cv2
import numpy as np
kernel = np.ones((15,15),np.uint8)
bg_subtractor = cv2.createBackgroundSubtractorKNN(history=500, dist2Threshold=400, detectShadows=True)
# Charger la vidéo
cap = cv2.VideoCapture(0)
# Création de l'objet Background Subtractor KNN
while True:
    ret, frame = cap.read()
    if not ret:
        break
    # Appliquer la soustraction de l'arrière-plan
    fg_mask = bg_subtractor.apply(frame)
    thresh = cv2.morphologyEx(fg_mask, cv2.MORPH_CLOSE, kernel)
    # Affichage des résultats
    cv2.imshow('Frame', frame)
    cv2.imshow('Foreground Mask', fg_mask)

    # Quitter sur appui de la touche 'q'
    if cv2.waitKey(30) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Comment choisir ces paramètres ?

1. Choix de history (taille de la mémoire) :

- **Petit history** (ex : 50-100 frames) : Si la scène change fréquemment ou si vous avez besoin de réagir rapidement aux changements (par exemple, des objets apparaissant soudainement).
- **Grand history** (ex : 500-1000 frames) : Si la scène est assez stable et que vous voulez une modélisation robuste, mais plus lente à réagir aux objets en mouvement.

2. Choix de `dist2Threshold` :

- **Petit `dist2Threshold`** (ex : 100-200) : Si vous avez besoin d'une détection très précise des objets en mouvement, avec une grande sensibilité aux petits changements dans la scène (mais potentiellement plus de bruit).
- **Grand `dist2Threshold`** (ex : 300-500) : Si la scène contient beaucoup de variations comme des ombres ou des changements de lumière, un seuil plus élevé peut aider à éviter les fausses détections.

3. Détection des ombres (`detectShadows`) :

- **Activé** : Si vous travaillez dans des environnements extérieurs ou sous éclairage variable, où les ombres pourraient affecter la détection des objets en mouvement.
- **Désactivé** : Si la scène est bien éclairée ou si les ombres ne posent pas des problèmes.