

# Compte Rendu

## TP1 : Chargement d'une vidéo avec OpenCV et Segmentation spatiale/temporelle

Enseignante : Dr. Rabaa YOUSSEF

Réaliser par : Jihen Fennani

Mariem Makni

Mariem Kilani

### Objectif du TP :

L'objectif de ce TP est de mettre en œuvre différentes techniques de **segmentation vidéo** pour détecter et analyser les objets en mouvement dans une séquence vidéo. À travers plusieurs méthodes telles que la **soustraction d'images**, la **différenciation de frames** et la **soustraction de fond adaptative**.

### I. Acquisition Chargement et Sauvegarde

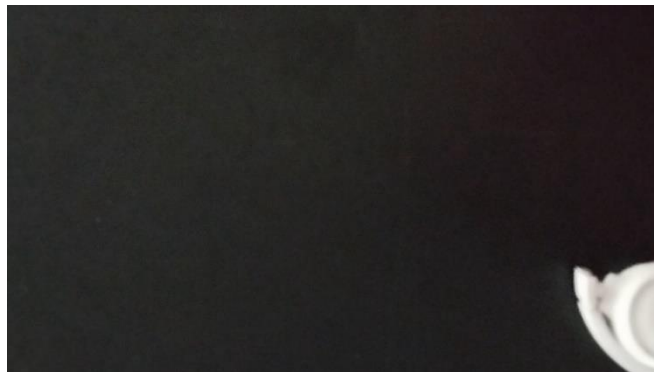
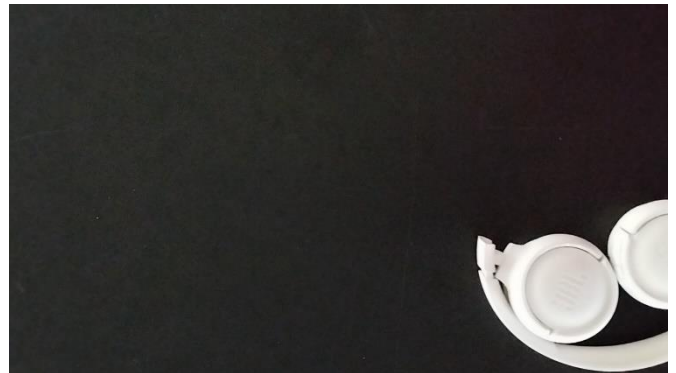
Dans cette première partie du TP, nous avons travaillé avec une vidéo d'une durée de 5 secondes, capturée à l'aide d'un smartphone Redmi Note 11. La vidéo a été enregistrée à un framerate adapté pour analyser les mouvements d'une scène simple. Voici le lien vers la vidéo utilisée : [\[le lien de vidéo\]](#).

Les informations extraites de la vidéo sont les suivantes :

```
In [29]: runfile('E:/RT5/Analyse vidéo/untitled7.py', wdir='E:/RT5/Analyse vidéo')
Informations sur la vidéo :
Largeur : 1920 pixels
Hauteur : 1080 pixels
Nombre total de frames : 144
FPS (frames per second) : 24.547593161525437
Frame 0 sauvegardée : frame\frame_0.jpg
Frame 30 sauvegardée : frame\frame_30.jpg
Frame 60 sauvegardée : frame\frame_60.jpg
Frame 90 sauvegardée : frame\frame_90.jpg
Frame 120 sauvegardée : frame\frame_120.jpg
Fin de la vidéo
```

Pour illustrer certaines étapes de notre analyse, nous avons sauvegardé des frames à intervalles réguliers, tous les 30 frames.

Les frames enregistrés sont les suivantes :



## II. Segmentation spatiale : Conversion et seuillage

Dans cette partie, nous allons appliquer des méthodes de **segmentation spatiale** pour analyser la vidéo frame par frame. L'objectif est de convertir chaque frame en niveaux de gris, puis d'appliquer différentes techniques de seuillage pour isoler les objets ou zones d'intérêt.

### 1. Conversion en niveaux de gris

La première étape consiste à **convertir chaque frame de la vidéo de l'espace de couleurs RGB en niveaux de gris**. Cette conversion est essentielle, car la majorité des méthodes de segmentation, y compris le seuillage, fonctionnent plus efficacement sur des images en niveaux de gris. Chaque frame de la vidéo est donc transformé en une image où l'intensité lumineuse de chaque pixel est représentée par une valeur unique de 0 (noir) à 255 (blanc).

### 2. L'algorithme de seuillage global heuristique

Dans le contexte de la segmentation d'images, un **algorithme de seuillage heuristique** consiste à utiliser une méthode empirique pour trouver un seuil qui permet de séparer les pixels en deux classes (noir ou blanc) sur la base des niveaux de gris. L'idée est d'ajuster un seuil de manière à obtenir la meilleure séparation entre les objets d'intérêt et l'arrière-plan.

```
In [20]: runfile('E:/RT5/Analyse vidéo/seuillage global heuristique.py', wdir='E:/RT5/Analyse vidéo')
Entrez le seuil initial T : 100
Lecture interrompue par l'utilisateur
Le seuil final T_final est : 99.69712522925894

In [21]: runfile('E:/RT5/Analyse vidéo/seuillage global heuristique.py', wdir='E:/RT5/Analyse vidéo')
Entrez le seuil initial T : 150
Fin de la vidéo
Le seuil final T_final est : 99.69712522925894

In [22]: runfile('E:/RT5/Analyse vidéo/seuillage global heuristique.py', wdir='E:/RT5/Analyse vidéo')
Entrez le seuil initial T : 120
Fin de la vidéo
Le seuil final T_final est : 99.72722443720214

In [23]: |
```

- Interprétation des résultats :

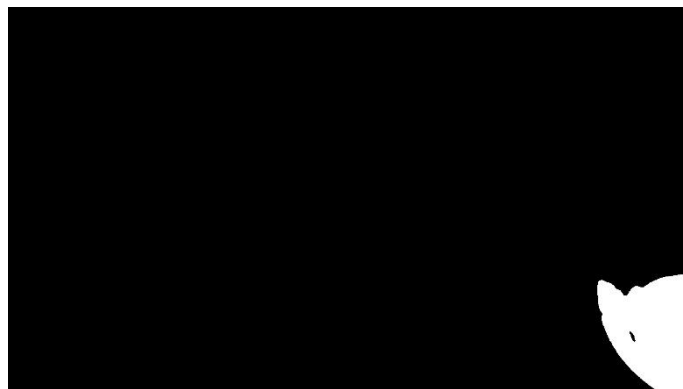
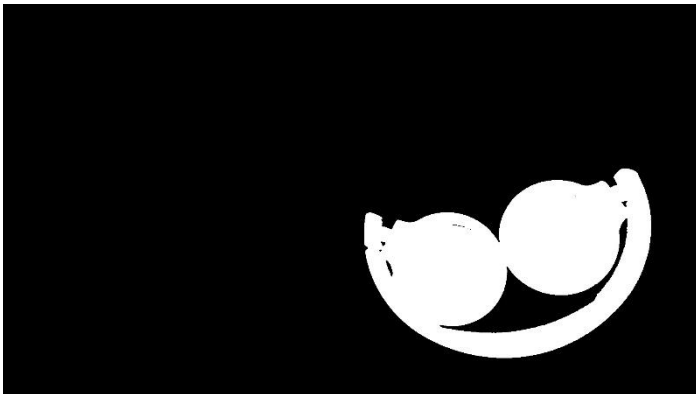
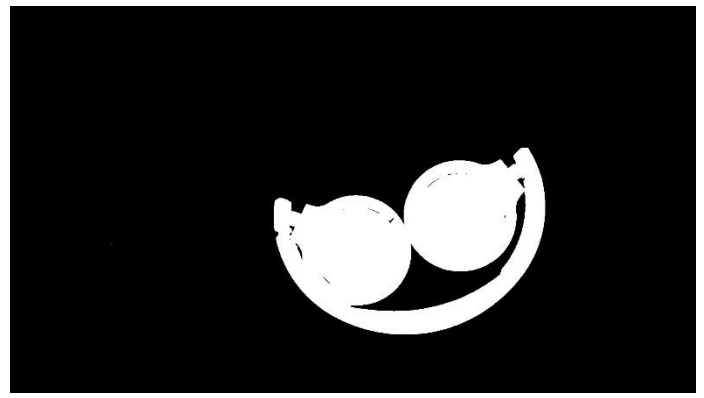
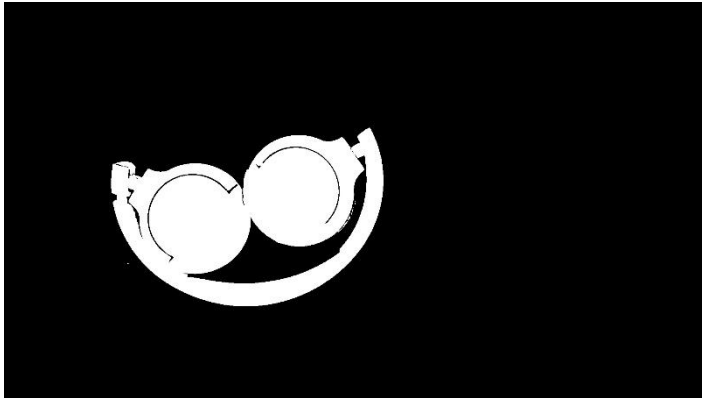
Lors de l'application de l'algorithme de **seuillage global heuristique**, plusieurs seuils initiaux T ont été testés pour la segmentation de la vidéo. Les résultats montrent une **convergence constante** vers un seuil T\_final proche de **99.7**, quelle que soit la valeur du seuil initial.

- Lorsque le seuil initial était fixé à **100, 120, ou 150**, l'algorithme a systématiquement convergé vers un seuil final proche de **99.7**, démontrant une **insensibilité au seuil initial**. Même avec des valeurs initiales éloignées du seuil optimal, l'algorithme ajuste rapidement le seuil pour atteindre une valeur cohérente.
- Cette **stabilité du seuil final** montre que l'algorithme est capable de trouver un seuil optimal qui segmente efficacement les objets d'intérêt de l'arrière-plan.

### 3. La méthode d'Otsu

Une autre méthode de segmentation appliquée est la **méthode d'Otsu**, qui permet de déterminer automatiquement le seuil optimal en analysant l'histogramme des niveaux de gris de l'image. Contrairement au seuillage heuristique qui nécessite un seuil initial, Otsu calcule directement le seuil qui minimise la variance intra-classe, c'est-à-dire qui sépare le mieux les pixels en deux classes distinctes (objet et fond).

Voici les frames résultant de la **segmentation par la méthode d'Otsu** :



- Interprétation des résultats :

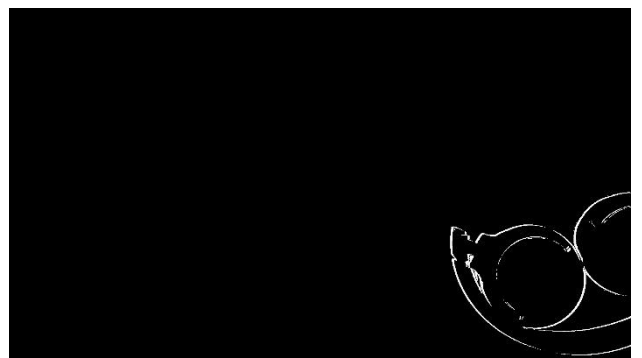
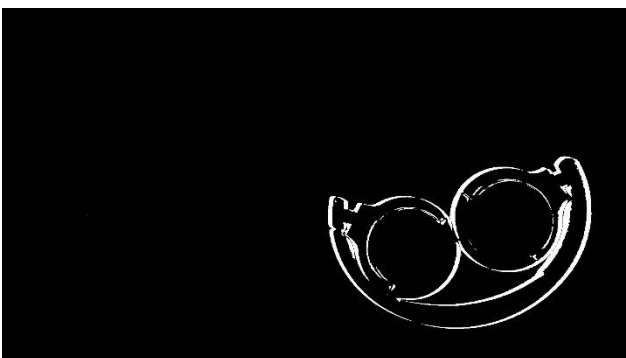
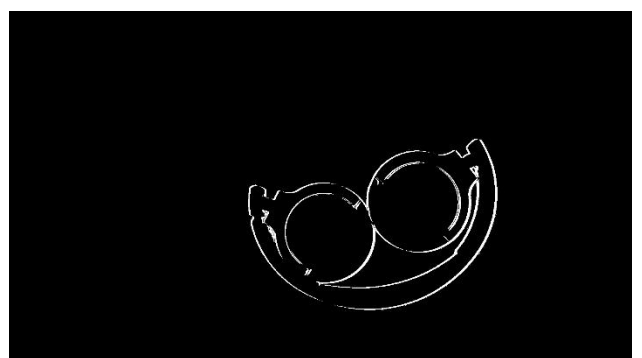
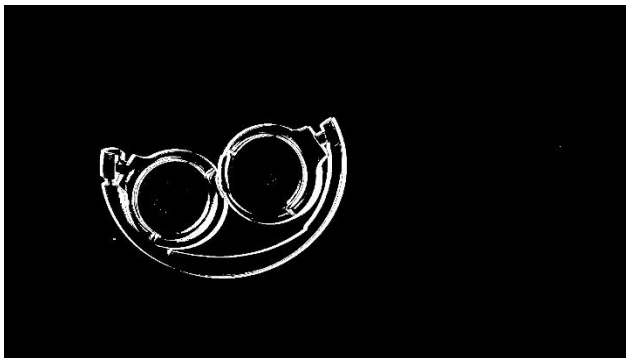
Les résultats obtenus avec la **méthode de seuillage d'Otsu** montrent une segmentation précise et fiable des objets d'intérêt tout au long des frames de la vidéo.

- Les objets apparaissent clairement en blanc, bien distincts du fond noir, démontrant l'efficacité du seuil calculé pour chaque frame.
- Les objets sont bien délimités, avec des contours nets.
- De petites imperfections apparaissent, notamment aux contours ou dans certaines zones plus complexes de l'image. Cela reste toutefois marginal et n'altère pas la qualité générale de la segmentation.

### III. Segmentation temporelle

#### 1. La méthode de soustraction de l'image $t-1$

Cette méthode consiste à soustraire chaque frame  $I(t)$  de la frame précédente  $B(t-1)$  pour détecter les objets en mouvement. Les différences significatives entre les deux frames représentent les objets qui ont changé de position, révélant ainsi les parties en mouvement dans la vidéo.

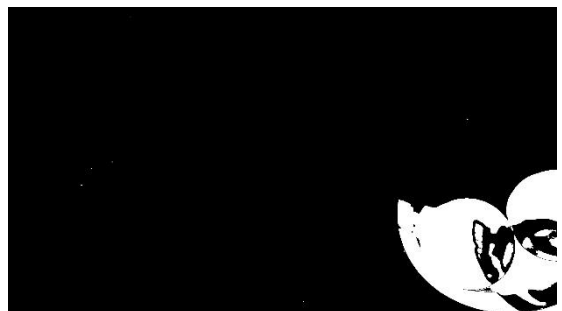
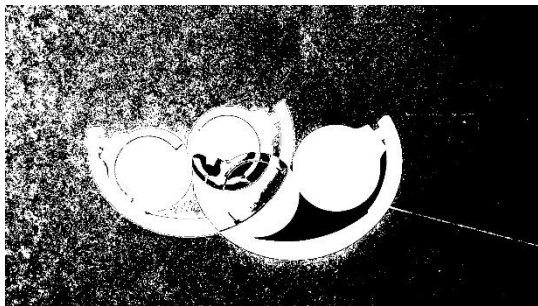




- Interprétation des résultats :
  - Les objets en mouvement sont clairement détectés, et leurs contours sont bien délimités. Les lignes blanches dans les images représentent les changements entre les frames, qui correspondent aux objets ayant bougé d'un frame à l'autre.
  - Les objets ne sont pas remplis, mais leurs contours sont bien tracés. Cela s'explique par le fait que la méthode de soustraction met principalement en évidence les bordures des objets qui ont changé d'une image à l'autre, tandis que les zones internes des objets restent similaires dans les frames successifs.

## 2. La méthode de soustraction de l'image t-1

La **méthode de différenciation de 3 frames** est une approche temporelle utilisée pour détecter les objets en mouvement dans une vidéo. Contrairement à la méthode de soustraction de l'image t-1, cette méthode compare non seulement la frame actuelle  $I(t)$  à la frame précédente  $I(t-1)$ , mais aussi une frame antérieure  $I(t-N)$  à la frame  $I(t-N-1)$  et une frame future  $I(t+N)$  à la frame  $I(t+N+1)$ , où  $N$  représente un nombre de frames d'écart.

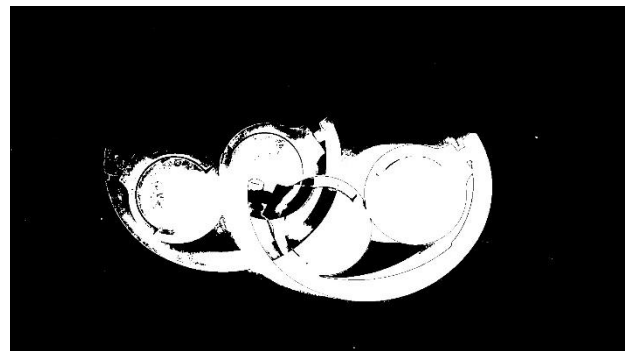
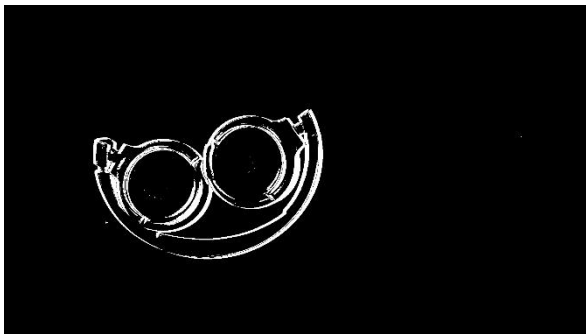




- Interprétation des résultats :
  - Les objets en mouvement sont clairement isolés, représentés par des zones blanches dans les images segmentées. Cette méthode capture bien les objets qui se déplacent dans la scène tout en réduisant les erreurs de segmentation liées aux objets statiques.
  - Les contours des objets en mouvement sont bien définis, ce qui permet une bonne visualisation des déplacements dans les frames successifs.
  - La **grande zone blanche** observée dans la frame 0 est causée par une **forte luminosité**, ce qui a temporairement affecté la détection des objets en mouvement et la segmentation.

### 3. La méthode Adaptive Background Subtraction

La **méthode Adaptive Background Subtraction** permet une mise à jour continue de l'arrière-plan et s'adapte mieux aux environnements dynamiques. Le choix du paramètre  $\alpha$  est crucial pour contrôler la vitesse d'adaptation. Une valeur trop faible rend l'arrière-plan trop statique, tandis qu'une valeur trop élevée risque d'intégrer des objets mobiles dans l'arrière-plan.







### ■ Interprétation des résultats :

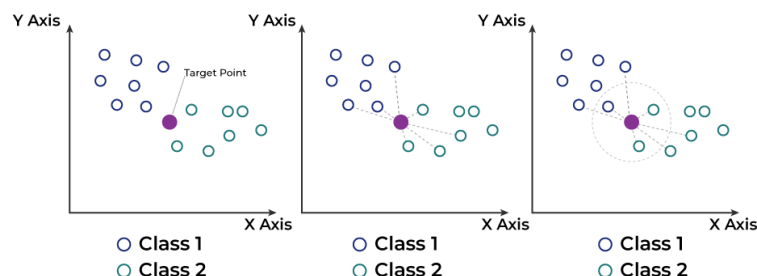
Les résultats obtenus pour la méthode de soustraction de fond adaptative avec un coefficient  $\alpha=0.05$  montrent que les changements dans l'image sont progressivement détectés au fil des frames. L'effet de  $\alpha$  ici est primordial :

- **$\alpha=0.05$**  : Cette valeur relativement basse signifie que l'image de fond est mise à jour lentement, ce qui permet de capturer des objets en mouvement tout en maintenant une distinction claire avec le fond. Cependant, si le mouvement est rapide ou si l'éclairage varie rapidement, cette valeur peut ne pas être suffisante pour bien adapter le fond.
- Un  $\alpha$  plus élevé rendrait l'adaptation plus rapide mais pourrait entraîner la perte d'objets en mouvement dans le fond.
- $\alpha$  plus faible favorise la détection continue des mouvements mais peut introduire des erreurs lorsque le fond change trop vite.

## 4. Autres méthodes

### 🌈 KNN (K-Nearest Neighbors)

La méthode KNN utilise un modèle de classification pour estimer l'arrière-plan. Pour chaque pixel, il garde en mémoire un certain nombre d'observations récentes. À partir de ces observations, il calcule la distance entre le pixel courant et les pixels observés dans le passé, ce qui lui permet de classer chaque pixel en tant qu'avant-plan ou arrière-plan en fonction de sa proximité avec les autres.



### Résultats et commentaires :

La méthode KNN est particulièrement efficace pour les environnements avec un fond plus dynamique. Elle est capable de mieux capturer les mouvements rapides et est plus robuste face aux changements abrupts dans l'environnement.