



## Master 1 Informatique

## Traitement d'image – 2024

## TD5 – Traitements d'images et de vidéos

**Le but** de ce TD est de :

- Développer une méthode de soustraction du fond.
- Implémenter l'algorithme d'incrustation d'une image dans une image du fond.
- Traiter des parties d'une image de la vidéo indépendamment.

### Exercice 1. Soustraction du fond d'une image

Utilisez les images `01_background.jpg` et `01_object.jpg` pour cet exercice.

Implémentez la méthode « Différence pixel à pixel ».

1. Récupérez les images `01_background.jpg` et `01_object.jpg`.
2. Implémentez la méthode de soustraction de fond.



### Exercice 2. Incrustation d'images

#### Simulation de transparence (*Alpha Blending*)

La simulation de transparence consiste à pouvoir combiner une image avec un arrière-plan, de façon à représenter une transparence partielle.

Pour combiner correctement des éléments d'image d'un arrière-plan, il faut associer et mémoriser un masque pour chaque élément. Ce masque contient les informations de recouvrement liées aux zones de l'image définies et aux zones vides.

**Math :** À chaque pixel de l'image, nous devons combiner la couleur de l'image du premier plan  $F$  et la couleur de l'image d'arrière-plan  $B$  en utilisant le masque alpha ( $\alpha$ ,  $0 \leq \alpha \leq 1$ ).

Pour obtenir l'image finale  $I = \alpha \times F + (1 - \alpha) \times B$

Si  $\alpha = 0$ , la couleur du pixel de sortie est prise dans l'image d'arrière-plan.

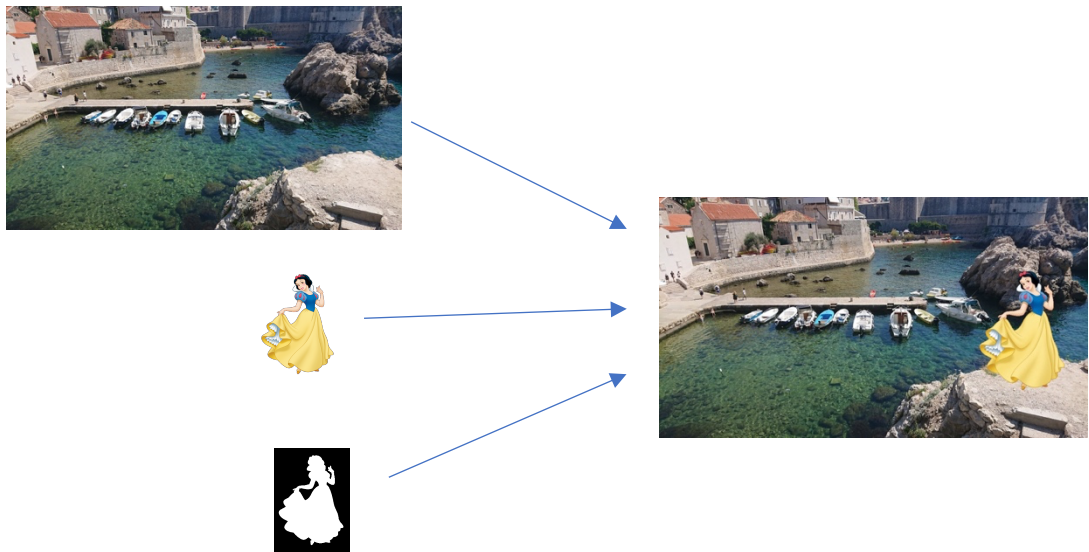
Si  $\alpha = 1$ , la couleur du pixel de sortie est prise dans l'image du premier plan.

Si  $0 < \alpha < 1$ , la couleur du pixel de sortie est le mix entre la couleur du pixel d'arrière-plan et du pixel de premier plan.

**Remarque :** les tailles des images **F** et **B** doivent être les mêmes.

Utilisez les images `02_background.jpg`, `02_personage.png` et `02_personage_alpha.png` pour cet exercice.

Écrivez un programme qui effectue une incrustation d'une image dans une autre image. Vous pouvez utiliser l'image `02_background.jpg` comme l'image d'arrière-plan et l'image `02_personage.png` comme l'image de premier-plan. L'image `02_personage_alpha.png` est un masque.



Les étapes à suivre :

1. Lire les images.
2. Convertir les valeurs uint8 en flottant.
3. Normaliser le masque `alpha` pour garder l'intensité entre 0 et 1.
4. Multiplier le premier plan avec le masque `alpha` normalisé.
5. Multiplier le fond avec  $(1 - \alpha)$ .
6. Ajouter le premier plan et l'arrière-plan masqués.
7. Afficher l'image.

## Lecture d'une vidéo

Utilisez la vidéo `video2.avi` et le fichier `video.py` pour cet exercice.

Téléchargez le fichier `video.py`. Le code dans ce fichier vous permet d'afficher la vidéo (`video2.avi`).

Avant l'exécution de ce code n'oubliez pas changer le chemin vers la vidéo.

### Exercice 3. Traitement des frames de vidéo

Utilisez la vidéo `video2.avi` et le fichier `frame_processing.py` pour cet exercice.

Récupérez le fichier `frame_processing.py` et testez ce code avec la vidéo `video2.avi`. Ce programme divise l'image en deux parties. Ensuite on affiche une partie de l'image en couleur et une autre partie de l'image en niveaux de gris.

Changez le code du fichier `frame_processing.py` en effectuant les tâches suivantes :

1. Divisez chaque image de la vidéo dans 6 parties.
2. Appliquez un traitement spécifique à chaque partie de l'image :
  - 1) Affichez une image sans changement.
  - 2) Affichez une image négative (utilisez le code de TD1).
  - 3) Affichez une image binaire (utilisez le code de TD1).
  - 4) Affichez seulement le canal vert d'image (utilisez le code de TD1).
  - 5) Affichez une image en appliquant le filtre sépia (voir Figure 1)

Pour le filtre sépia utilisez le noyau de transformation suivant :

$$\begin{pmatrix} 0.272 & 0.534 & 0.131 \\ 0.349 & 0.686 & 0.168 \\ 0.393 & 0.769 & 0.189 \end{pmatrix}$$

et appliquez ce noyau à l'image en utilisant la fonction `transform()` de OpenCV.

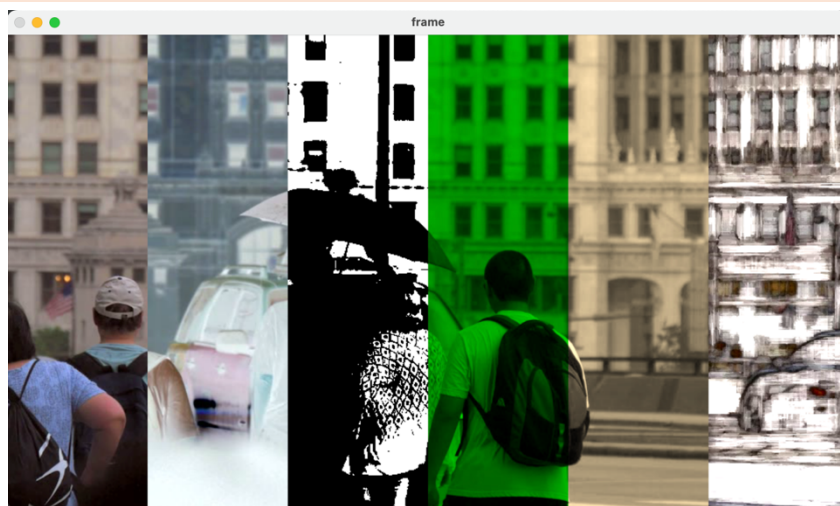


Figure 1 : Résultat obtenu.

- 6) Affichez une image qui ressemble à un dessin au crayon (voir Figure 1)

Utilisez la fonction `pencilSketch()` de OpenCV