

## TD7 Projet- Note d'intention

Nous avons mis en place un **système de filtre sur vidéo webcam en direct**. Pour la fenêtre et la gestion des Menu, nous avons utilisé l'interface graphique Tkinter. On peut désactiver ou (ré)activer les filtres dans le Menu nommé « *Choix des filtres* ».

**Le thème est « une femme des années 70 en vacances ».**

### Utilisation

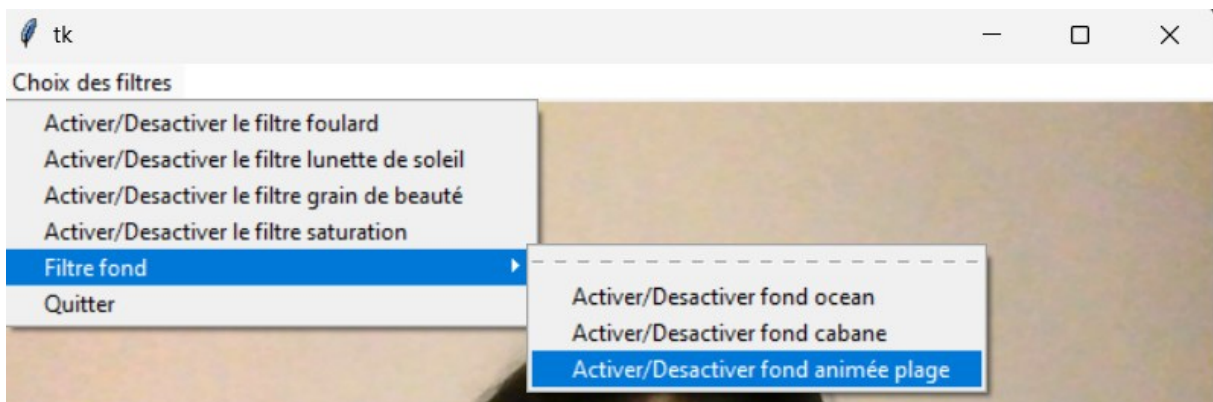
Placez la webcam sur une surface stable en face d'un fond blanc de préférence. Nous avons fait le choix de prendre le premier frame de la vidéo webcam comme fond de référence. Ainsi, lorsque vous lancez le programme, durant les 1ieres secondes d'enregistrement, ne vous mettez pas encore dans le champ de la caméra. Puis, vous pouvez vous positionner dans le champ et appliquer les différents filtres. Tous les filtres s'appliquent sur tous les visages détectés dans le champ.

Voici les filtres que l'on peut appliquer :

- filtre lunette : Des lunettes incrustées sur la vidéo et positionnés au niveau des 2 yeux.
- filtre foulard : Un foulard incrusté sur la vidéo et positionnés autour de la tête.
- filtre grain de beauté : Un grain de beauté incrusté sur la vidéo et positionnés au-dessus des lèvres.
- filtre saturation : Filtre appliquée sur toute l'image, saturation plus élevée, pour donner l'effet année 70.
- filtre fond : Un sous menu contenant différent arrière-plan

Remarque : *On ne peut utiliser qu'un fond à la fois, si l'un est activé, tous les autres sont désactivés par défaut.*

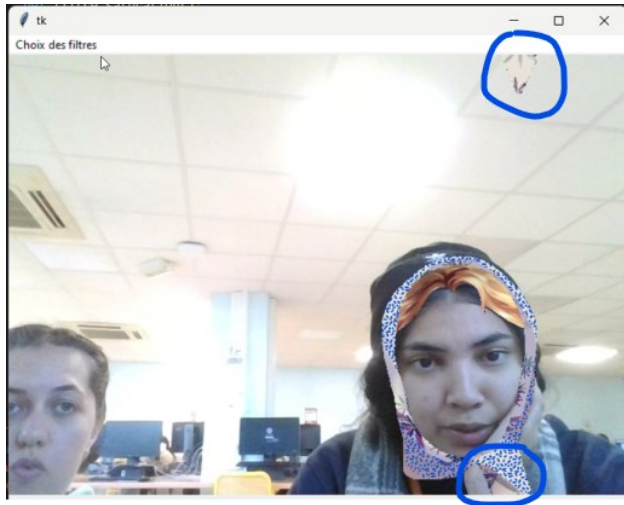
- fond océan : active un arrière-plan statique avec une feuille de palmier au premier plan
- fond cabane : active un arrière-plan statique avec une cabane sur l'eau
- fond animée plage : active un arrière-plan animé (des mouettes volant sur le haut de l'écran, animation sur 4 images seulement, fait par nous-même sur Procreate)



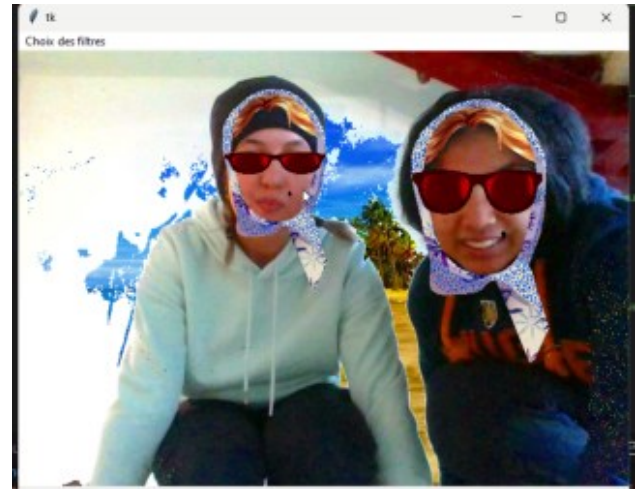
Pour quitter la fenêtre et arrêter le programme, vous pouvez aller dans *Choix des filtres>Quitter*.

## Toutes les images générées

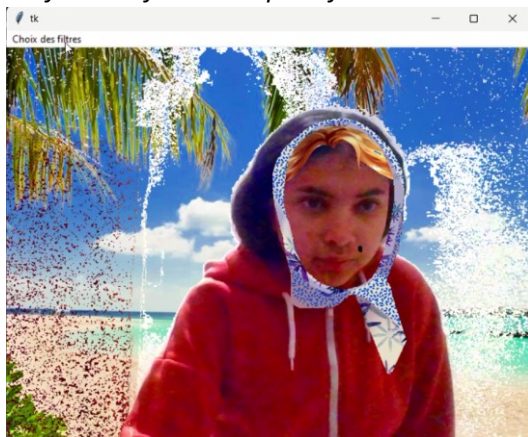
*Le filtre Foulard qui apparaît au bord opposé*



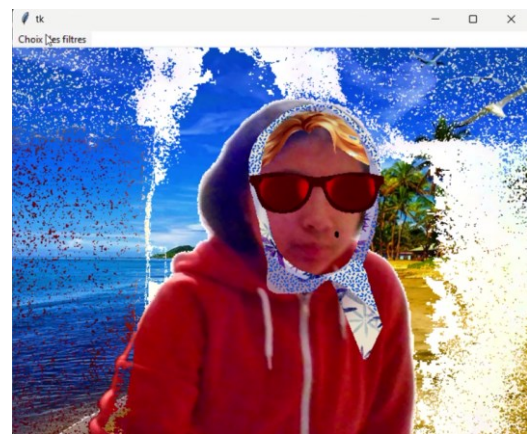
*Tous les filtres de détection par les cascades de Haar activées*



*Un filtre de fond statique + filtre contraste*



*Filtre du fond animé(en 4 images)*



## Les problèmes rencontrés et limites

### 1) Mise en place de la barre des Menu

Il était difficile de mettre en place la barre de menu sur la fenêtre « Frame » avec la vidéo, il y avait toujours 2 fenêtres différentes. On avait réussi à mettre une seule image de capture d'écran sur la fenêtre, mais pas la vidéo en entier.

Avec la fonction, « `master.after(10, update_image)` », on a pu mettre la vidéo en entier, plutôt que qu'une capture d'écran.

### 2) Activation et désactivation des filtres

Il fallait trouver un moyen d'optimiser le code lors de l'utilisation des filtres. Par exemple, quand on activait le filtre « lunette », la vidéo ralentissait drastiquement. Ainsi, il fallait que nous pensions à le désactiver. C'est pourquoi, nous avons mis en place les variables globales `bool_activate_filtre_lunette` ou même `bool_activate_filtre_saturation`. Lorsque l'on sélectionne « *Choix de filtre* » *Activer/Désactiver filtre lunette* », cela appelle simplement la fonction `filtre_lunette()` qui va mettre le booléen `bool_activate_filtre_lunette` en son booléen contraire. Et dans l'implémentation du code d'affichage des frames, nous entrons (ou non) dans la condition, `if bool_activate_filtre_lunette`.

```
def update_image():  
    ret, frame = cap.read()  
    if ret:  
        p1 = frame.copy()  
        # Gestion du filtre lunettes  
        if bool_activate_filtre_lunette: ...  
        # Gestion du filtre sépia  
        if bool_activate_filtre_sepia: ...
```

#### Limite :

La vidéo ralentit lorsqu'on utilise certains filtres. Il faudrait éteindre certains filtres pour le rendre plus fluides.

### 3) Positionnement des lunettes

Il était également difficile de bien positionner les lunettes sur les yeux. Il fallait que l'on calcule les coordonnées des lunettes en fonction de la position des yeux. On a dû décortiquer le vecteur « eyes » (le résultat de détection des yeux ) pour obtenir les dimensions exactes.

#### Limite :

A quelques fréquences, les lunettes disparaissent. Le filtre lunette ne s'active que lorsque l'on détecte les 2 yeux. Cependant, parfois, il n'en détecte qu'un seul. Aussi, lorsqu'on pivote légèrement notre tête, les lunettes ne pivotent pas avec nous.

### 4) Mise en place du filtre Foulard et du Grain de beauté

Pour le visage, on voulait mettre un filtre foulard et un grain de beauté. On a donc dessiné et fait son masque par la même occasion via le logiciel Procreate. Le plus dur était de calibrer le filtre sur la tête ou au-dessus de la bouche. On a donc modifié les paramètres jusqu'à ce que les filtres aient l'air cohérent.

Autre difficulté, lorsque des pixels de notre filtre foulard dépassait la zone d'image de la webcam, ça nous faisait une erreur « out of band ». Ainsi, pour contrer cela, on a fait le choix de faire apparaître la partie hors champs sur son bord opposé (traitement des bords : extension).

### 5) Calibrage de la détection de la bouche pour le grain de beauté

En ce qui concerne le filtre grain de beauté, le détecteur de bouche, appliqué sur la zone d'intérêt (donc le visage), était trop sensible. Il en détectait plus d'un à chaque fois, et parfois, il détectait les yeux. Ce qu'on a décidé de faire, c'est de réduire la zone d'intérêt à la partie inférieure du visage. Ainsi, cela réduisait le nombre de bouche détectée. Nous avons appliqué cette méthode sur les yeux, en augmentant la sensibilité de détection.

### 6) Lenteur du programme

Pour l'incrustation de l'image de fond, à l'origine, on voulait faire une double boucle qui parcourait toute l'image et incrustait l'arrière-plan. Cependant, cela ralentissait drastiquement le programme.

```
for i in range (0,frame.shape[0]):  
    for j in range (0,frame.shape[1]):  
        if (capture_d_image_de_fond[i,j,0] == frame[i,j,0]):  
            frame[i,j]=fond_frame_resize[i,j]
```

Pour le remplacer et pour l'optimiser, nous avons utilisé l'opérateur par défaut '==' l'égalité des images :

```
# Incrustation de l'animation
mask = (capture_d_image_de_fond == frame)
frame[mask] = animation_frame_resize[mask]
```

Ainsi, le traitement est redevenu fluide.

#### 7) Calibrage de la détection d'arrière-plan

Au début, lorsqu'on testait le filtre arrière-plan sur différents fonds (une fois, on a testé sur un fond blanc puis sur un fond bleu), le filtre ne s'activait pas totalement (juste quelques bruits sur les coins de la vidéo). On constatait que ça fonctionnait surtout pour les fonds blancs. On a regardé notre code et on a constaté que dans notre condition d'incrustation d'image, on a posé une égalité parfaite entre le plan actuel et le plan de fond. Mais aussi, on examinait seulement le canal rouge (le '2' sur image ci-dessous). C'est pour cela que le filtre arrière-plan pouvait varier.

```
mask = (frame[:, :, 2] == capture_d_image_de_fond[:, :, 2])
```

Ainsi, au lieu d'avoir une condition d'égalité parfaite, on a mis une condition sur une intervalle d'écart type (ex : intensité 20). Enfin, au lieu d'analyser le canal rouge uniquement, on analyse plutôt sur le niveau de gris.

```
gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
gray_capture_d_image_de_fond = cv2.cvtColor(capture_d_image_de_fond, cv2.COLOR_BGR2GRAY)
mask = ((gray_frame - 20 <= gray_capture_d_image_de_fond) & (gray_capture_d_image_de_fond <= gray_frame + 20))
```

### Contraintes et limites générales

Comme on peut s'en douter, il y a des limites à l'utilisation de ses filtres. On va les énumérer :

- ❖ Il faut poser la caméra webcam sur une surface stable pour avoir toujours le même fond.
- ❖ Il faut trouver un fond de préférence blanc pour bien incruster l'arrière-plan et éviter le plus que possible les variations de luminosité. Si ce n'est pas respecté, il se peut que l'arrière-plan ne soit pas appliqué et que vous voyez seulement des bruits sur l'image.
- ❖ Durant les premières secondes d'enregistrement de la vidéo, ne pas se mettre dans le champ de la vidéo.
- ❖ Quand on active le filtre lunette, le programme ralentit à cause du calcul de positionnement de l'image.
- ❖ Le programme est lent dès que l'on active tous les filtres en même temps.
- ❖ Pour la détection des yeux, ne pas mettre de lunette, de plus, ayant moi-même légèrement les yeux bridés (*Finaritra*), parfois le capteur de yeux ne les détecte pas.
- ❖ Certains paramètres dans le code sont encore à calibrer si l'on veut de bon résultat. Cela prendrait plus de temps si l'on veut automatiser cela. Par exemple :
  - L'arrière-plan choisit (le fond, la luminosité, de nuit ou de jour).
    - Ex : l'écart type de l'intervalle de détection de l'arrière-plan (voir problème n°6 *Calibrage de la détection d'arrière-plan* ci-dessus)
  - La sensibilité des détecteurs sur le visage (voir problème n°5 *Calibrage de la détection de la bouche pour le grain de beauté*).