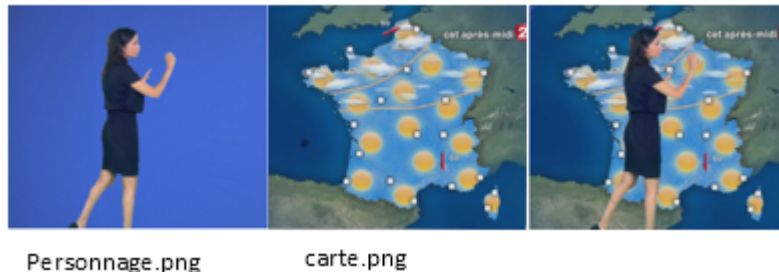


Projet3e : météo (dd7a-3798613)

Les émissions de télévision utilisent couramment les incrustations d'images pour créer des arrières plans-dynamiques. C'est en particulier le cas pour la présentation des prévisions météo.



Le principe de cette incrustation consiste d'abord à connaître la valeur typique $[R,V,B]$ des pixels « bleus » de l'arrière plan afin de distinguer les pixels associés au personnage des pixels associés à l'arrière plan. La technique consiste ensuite à incruster les pixels du personnage sur la photo de la carte météo. On donne ci-dessous les lignes de code permettant d'obtenir les listes et les tableaux numpy des photos Personnage.png et carte.png

```
In [3]: 1 from PIL import Image
2 import numpy as np
3 tab_p=Image.open("C:/Users/User/Documents/travail/TSI2_2024_2025/in
4 tab_p=np.array(tab_p,dtype="uint8")
5 liste_p=tab_p.tolist()
6 tab_c=Image.open("C:/Users/User/Documents/travail/TSI2_2024_2025/in
7 tab_c=np.array(tab_c,dtype="uint8")
8 liste_c=tab_c.tolist()
9
```

1) Analyser les pixels « bleus » de l'arrière-plan $tab_p[100,100]$, $tab_p[95,100]$, $tab_p[100,95]$, $tab_p[105,100]$ et $tab_p[100,105]$. L'arrière-plan est-il uniforme ?

Le fond n'est pas rigoureusement uniforme : il y a une fluctuation du niveau R,V,B

2) Dans la suite, on définit un pixel de référence $[R,V,B]$:

version avec des listes

3) En utilisant une boucle for, construire une fonction $distance(liste,i,j,pixel_ref)$ qui prend pour argument une liste associée à une photo et qui renvoie la norme du vecteur mesurant l'écart entre la couleur du pixel $liste[i,j]$ considéré avec celle d'un pixel référence nommé $pixel_ref$.

4) Définir une fonction $incrustation(liste_p,list_c,condition)$ qui prend pour argument deux listes ainsi qu'un entier ($condition$). Cette fonction permettra, en parcourant les pixels de $liste_p$ à l'aide de deux boucles for, d'incruster le personnage s'il s'agit d'un pixel différent du fond

5) Mesurer le temps nécessaire pour réaliser l'incrustation d'image avec des listes et obtenir cette image avec condition=50

Version avec des tableaux numpy

6) En prenant encore comme référence le pixel bleu aux coordonnées (100,100), obtenir un tableau mask ayant le même nombre de lignes et de colonnes que tab_p, et dont les éléments sont faux s'il s'agit d'un pixel bleu du fond et vrais s'il s'agit d'un autre pixel. On veillera à éviter les problèmes d'overflow en travaillant avec des int64!

7) Proposer alors une expression simple permettant d'incruster l'image du personnage sur la carte météo. Mesurer le temps nécessaire pour faire cette incrustation et préciser si ce calcul peut se faire en temps réel.