
École Centrale de Lyon 2015-2016
INFO : tc1
Rapport BE2 : L'algorithme Quadripartition appliqué pour la segmentation

KONAN Jordan N'guessan Ziahi

NDIAYE Serigne Fallou

Groupe TD : A2a

Partie I : Split

→ **nom du fichier : split.py**

Pour la réalisation de l'algorithme split, on a utilisé au total 5 fonctions y compris la fonction Quadripartition elle même.

Pour le seuil du split on a choisi $s=10$ et le temps d'exécution avec l'image « Image8.bmp » est de : $t=22,79s$

----- Résultat exécution -----



Après Split



Avant Split

PS : La fonction Lirepixel() n'a pas été utilisée dans le programme split

Partie II : Merge

→ **Nom du fichier : traitementImage.py**

→ **Fichiers associés : adjacence.py et effaceDoublon.py**

La réalisation du programme de Fusion (Merge) s'appuie sur le programme Split décrit plus haut. Ainsi, la fonction « esthomogene » a été modifiée pour obtenir L_regions. En plus de cela, on a utilisé 5 autres fonctions dans le fichier principal et certaines fonctions des fichiers qu'on a inclus.

Il faut noter par contre que pour la fonction «estAdjacente » qui teste l'adjacence de deux régions on a pensé qu'il est mieux de diviser chaque région en quatre segments, puis de voir s'il y a intersection entre les segments des deux régions. En effet, cette façon de faire nous assure que tous les cas seront pris en compte comme par exemple celui présenté ci-après (qui serait un cas exceptionnel si on avait travaillé avec les points qui définissent les régions)

zone1	zone3
zone2	

Le temps d'exécution du programme avec un seuil de merge (seuil=30) est de $t = 496s$



Image initiale

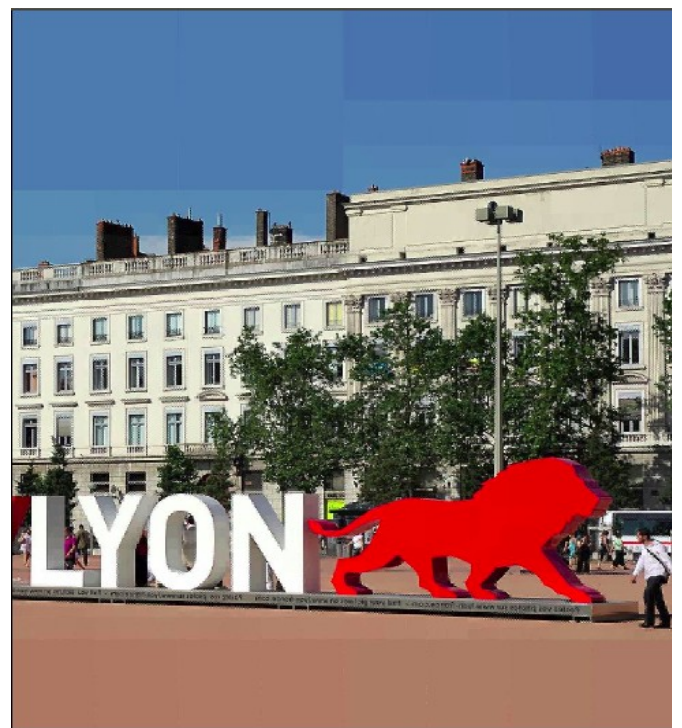


Image après Split

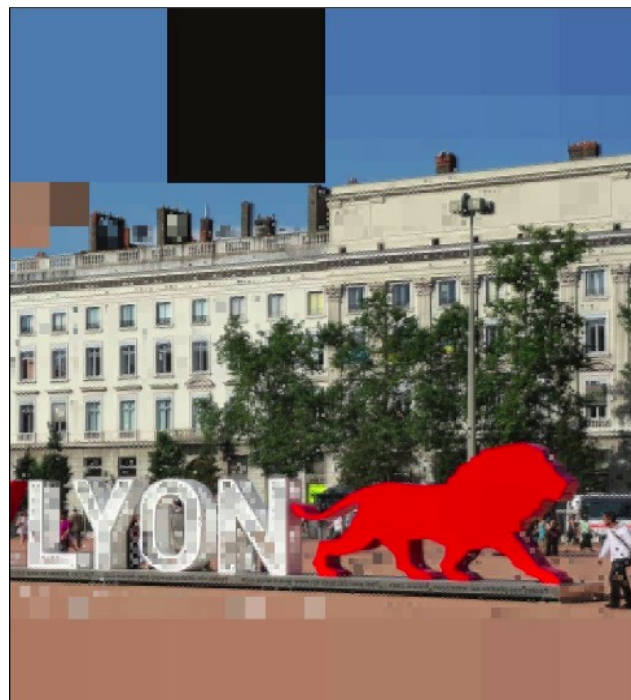


Image après Merge