



Analyse et traitement d'images

Compte-rendu TP6

Segmentation d'une image par Split and Merge

Louis Jean
Master 1 IMAGINE
Université de Montpellier
N° étudiant : 21914083

8 mars 2024

Table des matières

1	Introduction	2
2	Division d'une image en 4 régions	3
3	Étape de division récursive	4
4	Étape de fusion	7
5	Extension aux images couleur	8
6	Conclusion	10

1 Introduction

L'objectif de ce TP est de comprendre la segmentation d'une image avec une approche basée région : split and merge. Pour ce TP, j'ai choisi une image couleur *johnwick.webp*, de taille 512x512 pixels (il est important que la taille de l'image utilisée soit une puissance de 2 dans ce TP), que j'ai ensuite transformée au format ppm et pgm.



(a) johnwick.ppm



(b) johnwick.pgm

FIGURE 1 – johnwick.ppm et johnwick.pgm

2 Division d'une image en 4 régions

Dans un premier temps, j'ai divisé mon image en 4 régions distinctes. Pour chacune de ces régions, j'ai affecté la valeur moyenne de cette région à tous ses pixels, et calculé la variance et l'écart-type.

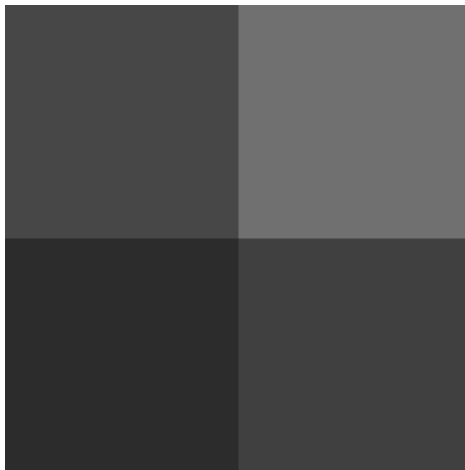


FIGURE 2 – johnwick.pgm divisée en 4 régions

On observe ces résultats (on lit les blocs de haut en bas puis de gauche à droite) :

- Bloc 0 : $\bar{x} = 71.9986$, $\sigma^2 = 3041.18$, $\sigma = 55.1469$
- Bloc 1 : $\bar{x} = 44.1525$, $\sigma^2 = 2177.64$, $\sigma = 46.6652$
- Bloc 2 : $\bar{x} = 112.995$, $\sigma^2 = 2900.64$, $\sigma = 53.8575$
- Bloc 3 : $\bar{x} = 64.0715$, $\sigma^2 = 2654.98$, $\sigma = 51.5265$

avec \bar{x} la moyenne, σ^2 la variance et σ l'écart-type.

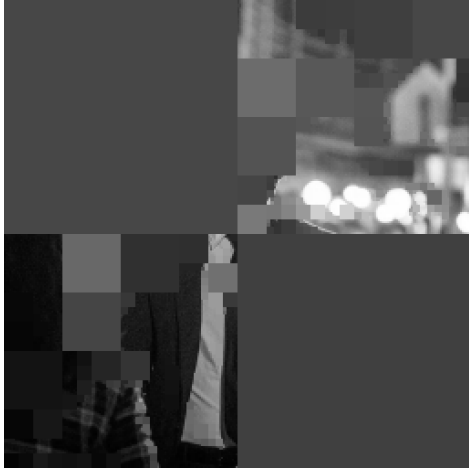
3 Étape de division récursive

Pour améliorer cette division et obtenir des régions plus homogènes, on propose de diviser récursivement chaque région en 4 sous-blocs (c'est-à-dire implémenter un quadtree) jusqu'à atteindre le seuil voulu ou la taille minimale de bloc choisie. L'algorithme de cette division récursive est donc :

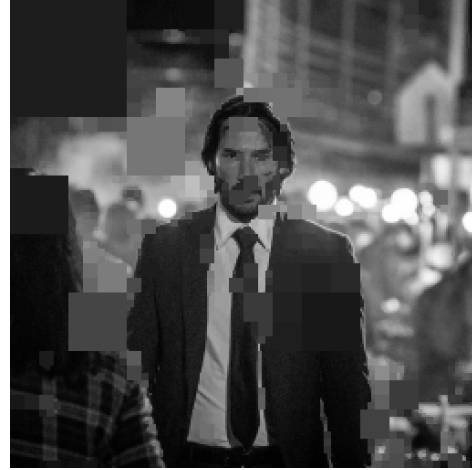
Algorithme 1 Division récursive d'une image

```
1: Fonction DIVISIONRECURSIVE(ImgIn, ImgOut, x_debut, x_fin, y_debut,  
   y_fin, seuil, nW)  
2:   largeur  $\leftarrow x_{fin} - x_{debut}$   
3:   hauteur  $\leftarrow y_{fin} - y_{debut}$   
4:   taille  $\leftarrow largeur \times hauteur$   
5:   moyenne  $\leftarrow moyenne\_bloc(ImgIn, x_{debut}, y_{debut}, x_{fin}, y_{fin}, nW)$   
6:   variance  $\leftarrow variance\_bloc(ImgIn, x_{debut}, y_{debut}, x_{fin}, y_{fin}, nW)$   
7:   ecart_type  $\leftarrow \sqrt{variance}$   
8:   if taille  $\leq 4$  or ecart_type  $\leq seuil$  then  
9:     for y  $\leftarrow y_{debut}$  to yfin - 1 do  
10:      for x  $\leftarrow x_{debut}$  to xfin - 1 do  
11:        ImgOut[y  $\times$  nW + x]  $\leftarrow moyenne$   
12:      end for  
13:    end for  
14:  else  
15:    x_milieu  $\leftarrow (x_{debut} + x_{fin})/2$   
16:    y_milieu  $\leftarrow (y_{debut} + y_{fin})/2$   
17:    DIVISIONRECURSIVE(ImgIn, ImgOut, x_debut, x_milieu, y_debut,  
      y_milieu, seuil, nW)  
18:    DIVISIONRECURSIVE(ImgIn, ImgOut, x_milieu, x_fin, y_debut, y_milieu,  
      seuil, nW)  
19:    DIVISIONRECURSIVE(ImgIn, ImgOut, x_debut, x_milieu, y_milieu, y_fin,  
      seuil, nW)  
20:    DIVISIONRECURSIVE(ImgIn, ImgOut, x_milieu, x_fin, y_milieu, y_fin,  
      seuil, nW)  
21:  end if  
22: end Fonction
```

J'ai décidé de prendre une taille minimale de bloc $B = 4$ (donc 2×2). Après avoir implémenté cet algorithme, voici les résultats que l'on peut obtenir pour différents seuils S .



(a) $S = 57.6337$ (écart-type de l'image globale)



(b) $S = 30$



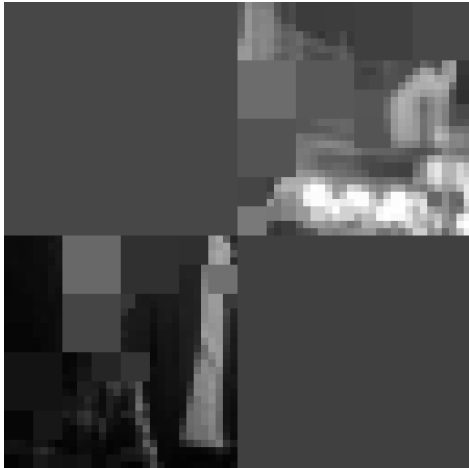
(c) $S = 15$



(d) $S = 10$

FIGURE 3 – Images subdivisées récursivement avec $B = 4$

En prenant maintenant une taille minimale de bloc $B = 64$ (donc 8×8), voici ce que l'on obtient.



(a) $S = 57.6337$ (écart-type de l'image globale)



(b) $S = 30$



(c) $S = 15$



(d) $S = 10$

FIGURE 4 – Images subdivisées récursivement avec $B = 64$

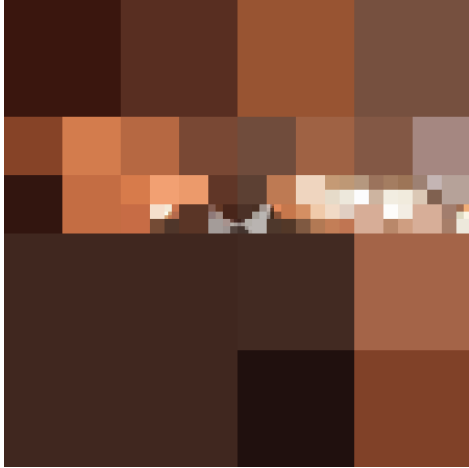
On remarque que des plus gros blocs donnent un résultat similaire pour des hautes valeurs de seuil, mais sont sensiblement différentes pour des faibles valeurs de seuil.

4 Étape de fusion

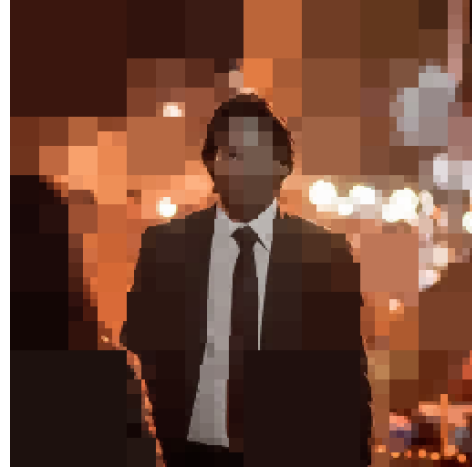
Par manque de temps, je n'ai pas pu implémenter la structure de données du RAG ni l'algorithme de fusion. J'essaierai tout de même de faire cela chez moi afin de bien terminer ce TP. Dans le laps de temps imparti, j'ai préféré me concentrer sur les images couleur.

5 Extension aux images couleur

Par analogie, on peut étendre les méthodes vues ci-dessus aux images couleur.



(a) $S = 50$



(b) $S = 30$



(c) $S = 15$



(d) $S = 10$

FIGURE 5 – Images subdivisées récursivement avec $B = 4$

En prenant $B = 64$, voici les résultats.



(a) $S = 50$



(b) $S = 30$



(c) $S = 15$



(d) $S = 10$

FIGURE 6 – Images subdivisées récursivement avec $B = 64$

On fait les mêmes observations que pour les images en niveaux de gris.

6 Conclusion

J'ai pris du plaisir à réaliser ce TP même si j'aurai aimé avoir plus de temps pour pouvoir appréhender et implémenter la fusion. Je pense que mon erreur a été de me lancer dans la division récursive sans même penser à l'étape de fusion, ce qui fait que je n'ai pas du tout codé en ayant en tête que je devrais plus tard rajouter une nouvelle structure de données et l'ajouter à mon code existant. En ayant regardé des exemples de Split and Merge sur internet ([ici](#)), j'ai l'impression que cette technique est très convaincante.