

Reconnaissance des formes

TP5 : SEGMENTATION AUTOMATIQUE PAR ANALYSE D'HISTOGRAMME

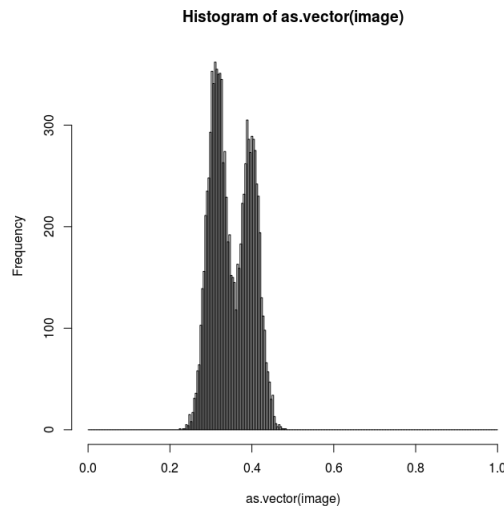
Par

Douaille Erwan & Francois Remy

Introduction

Trouver le seuil de binarisation est une tâche peu évidente et jusque aujourd'hui nous la déterminions visuellement et, ou par méthode empirique. Comment trouver ce seuil de façon automatisée et le plus optimisé possible ? C'est ce que nous allons expérimenter dans ce compte rendu.

Question 1

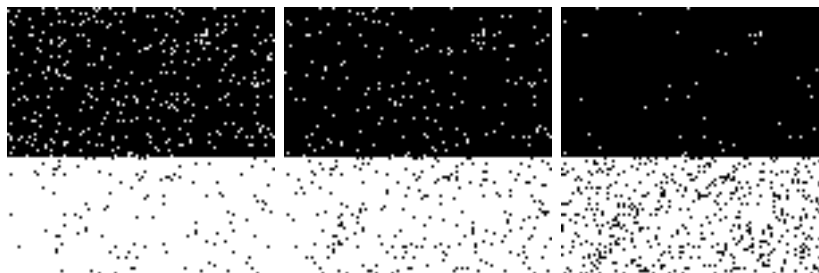


Les valeurs de l'histogramme représentent le nombre de pixels pour un niveau de gris donné. Les valeurs données (les seuils) représentent le seuil de coupure de notre histogramme, c'est un pourcentage. Notre histogramme va de 0 à 255 où de 0 à 1. 0.35, définit le seuil où on sépare l'histogramme.

```

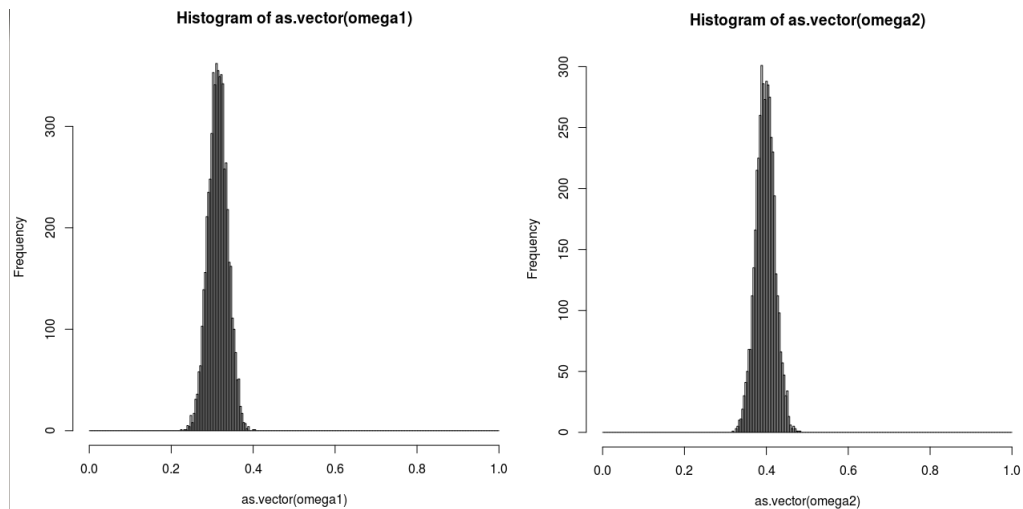
1 binaire1 <- (image - 0.35)>= 0
2 binaire2 <- (image - 0.36)>= 0
3 binaire3 <- (image - 0.37)>= 0
4 display(binaire1, "binaire 1")
5 display(binaire2, "binaire 2")
6 display(binaire3, "binaire 3")
7 nbins <- 256
8 h <- hist (as.vector (image), breaks = seq (0, 1, 1 / nbins))

```



À la vue de nos binarisations on observe que aucune de ces images permet une parfaite binarisation. Soit la partie noir s'approche de la perfection, soit c'est la partie blanche.

Question 2



```

1 h1 <- hist (as.vector (omega1), breaks = seq (0, 1, 1 / nbins))
2 h2 <- hist (as.vector (omega2), breaks = seq (0, 1, 1 / nbins))
3 N = sum(h$counts) # Nb de pixels total de l'image
4 N1 = sum(h1$counts) # Nb de pixels de la classe omega 1
5 N2 = sum(h2$counts) # Nb de pixels de la classe omega 2
6 Pomega1 = N1/N
7 Pomega2 = N2/N

```

La probabilité pour un pixel d'être dans une classe particulière est :

- $P(\omega_1) = 0.56$
- $P(\omega_2) = 0.44$

Question 3

```
1 ndg <- 89
2 PIconditionnelle <- h$counts[ndg]/N
3 PO1conditionnelle <- h1$counts[ndg]/N1
4 PO2conditionnelle <- h2$counts[ndg]/N2
```

Nombre de pixels ayant le niveau de gris 89 dans chaque image sont :

- $I = 192$
- $\omega_1 = 162$
- $\omega_2 = 30$

Les probabilités d'avoir le niveau de gris 89 dans chaque image sont :

- $P(89/I) = 0.0192$
- $P(89/\omega_1) = 0.0289$
- $P(89/\omega_2) = 0.0068$

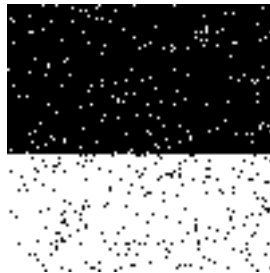
Question 4

```

1 # Seuillage automatique – Proba d’erreur
2 somme1 = 0:255
3 somme2 = 0:255
4 erreur = 0:255
5 # recherche du minimum
6 minimum_erreur = 1;
7 seuil_minimum_erreur = 0;
8
9 for (X in 1:255) {
10   somme1[X+1]=sum(h1$density[(X+1):256])/sum(h1$density[1:256])
11   somme1[X+1]=somme1[X+1]*Pomega1
12
13   somme2[X+1]=sum(h2$density[1:(X+1)])/sum(h2$density[1:256])
14   somme2[X+1]=somme2[X+1]*Pomega2
15
16   erreur[X+1] = somme1[X+1] + somme2[X+1]
17 # seuil correspondant a l’erreur minimale
18 if (erreur[X+1] < minimum_erreur )
19   seuil_minimum_erreur = X
20 if (erreur[X+1] < minimum_erreur )
21   minimum_erreur = erreur[X+1]
22 }
23
24 #seuillage de l’image
25 seuil_auto_img <- (image - seuil_minimum_erreur/255)>= 0##
26 display(seuil_auto_img)

```

On trouve un seuil optimal de 92. Le taux d’erreur de classification est de 0.0471. Voici l’image obtenue :



Question 5

Pour cette question nous avons appliqué tout ce qui a été vu précédemment. Concernant le code nous avons réutilisé l'intégralité des questions précédentes et changé le nom des images.



Conclusion

En conclusion on obtient une binarisation avec un seuil déterminé automatiquement par des calculs de probabilité, solution optimale. On a pu distinguer toutes les zones avec des seuils différents pour chaque zone.