

Master 1 Informatique

TP 5 RDF

Compte rendu de TP

Antoine CANDIA
Arthur D'Azemar

*Version 1.0
01 février 2017*

Table des matières

1	Question 1	4
2	Question 2	6
3	Question 3	7
4	Question 4	9
5	Question 5	11
6	Question 5-bis	12

Chapitre 1

Question 1

Les valeurs de l'histogramme représente la densité de pixel pour chaque nuance de gris de l'image en question. Il y a donc 256 densités différentes de représenté.

Les valeurs que l'on considère ici sont des valeurs pouvant servir de seuil potentiel de coupe de l'image en deux. On retrouve bien sur l'histogramme deux ensembles distincts de pixels et la valeur seuil se trouve approximativement à la jonction des deux, c'est la limite entre les deux.

Quand on regarde les résultats des images avec un seuil de 0.5, 0.55 et 0.6, on remarque que le 0.55 est à priori le plus semblable alors que les deux autres sont trop marqué sur l'un ou l'autre des parties de l'image. L'histogramme semble aller aussi dans ce sens quand car la jointure entre les deux ensembles est plus proche de 0.55 que 0.5 ou 0.6.

Le code R correspondant :

```
# Chargement de l'image
nom <- "2classes_100_100_8bits_2016.png"
image <- rdfReadGreyImage (nom)

# Calcul et affichage de son histogramme
nbins <- 256
h <- hist (as.vector (image), freq=FALSE, breaks = seq (0, 1, 1 / nbins))

# Segmentation par binarisation 0.5
seuil <- 0.50
binaire50 <- (image - seuil) >= 0

# Segmentation par binarisation 0.55
seuil <- 0.55
binaire55 <- (image - seuil) >= 0

# Segmentation par binarisation 0.6
seuil <- 0.6
binaire60 <- (image - seuil) >= 0
```

```
# Affichage des images binarisées
display (binaire50, "image binaire 0.50")
display (binaire55, "image binaire 0.55")
display (binaire60, "image binaire 0.6")
```

Les images créées :



Chapitre 2

Question 2

La probabilité pour un pixel d'être dans la classe de ω_1 est de 0.57 et donc de $1-0.57 = 0.43$ pour ω_2 (valeur également trouvé par le calcul en R).

Le code R correspondant est :

```
# Calcul des probas a priori des classes

p_omega1= sum(h1$counts[0:255])/ sum(h$counts[0:255])

p_omega2= sum(h2$counts[0:255])/ sum(h$counts[0:255])

#affichage des probabilités p_omega1 et p_omega2

print(p_omega1)
print(p_omega2)
```

Chapitre 3

Question 3

Il y a pour chaque image le nombre de pixel de niveau de gris 141 suivant :

- Image I : 118
- Image Omega1 : 51
- Image Omega2 : 67

Les probabilités pour un pixel de niveau de gris 141 d’être dans les images correspondante aux classes associés sont les suivantes :

- $P(141/I) = 0.0118$
- $P(141/Omega1) = 0.008947368$
- $P(141/Omega2) = 0.0155814$

Le code R correspondant :

```
# Calcul des probas conditionnelles
niveauGris <- 142

#Somme pixel de niveau de gris X selon l'image

nbPixelI <- h$counts[niveauGris]
nbPixel01 <- h1$counts[niveauGris]
nbPixel02 <- h2$counts[niveauGris]

# P(X/classe) = total(X)/total(Pixel image)
p_cond_image <- nbPixelI / sum(h$counts)
p_cond_omega1 <- nbPixel01 / sum(h1$counts)
p_cond_omega2 <- nbPixel02 / sum(h2$counts)

print(nbPixelI)
print(p_cond_image)

print(nbPixel01)
print(p_cond_omega1)
```

```
print(nbPixel02)
print(p_cond_omega2)
```


Chapitre 4

Question 4

Le seuil calculé est 0.545.
Le taux d'erreur est de 0.0481.

Le code R correspondant est :

```
# pour le seuil X calcul de l'erreur d'assignation
somme1 = 0:255
somme2 = 0:255
erreur = 0:255
# recherche du minimum
minimum_erreur = 1;
seuil_minimum_erreur = 0;

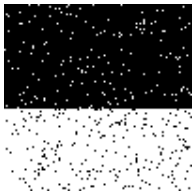
for (X in 1:255)
{
  somme1[X+1]=sum(h1$density[(X+1):256])/sum(h1$density[1:256])
  somme1[X+1]=somme1[X+1]*p_omega1

  somme2[X+1]=sum(h2$density[1:(X+1)])/sum(h2$density[1:256])
  somme2[X+1]=somme2[X+1]*p_omega2

  erreur[X+1] = somme1[X+1] + somme2[X+1]

# seuil correspondant a l'erreur minimale
  if (erreur[X+1] < minimum_erreur ) seuil_minimum_erreur = X
  if (erreur[X+1] < minimum_erreur ) minimum_erreur = erreur[X+1]
}

seuil = seuil_minimum_erreur/255
print(seuil_minimum_erreur)
print(seuil)
```



Chapitre 5

Question 5

La seule modification a été de modifier le nom des fichiers et relancer les calculs de seuillage automatique créée lors des questions précédentes.

On obtient un seuil à environ 0.5569 et un taux d'erreur de 0.8125.

On obtient l'image suivante :



En reprenant le seuil trouvé sur cette image et en l'appliquant à l'image du chiffre 1 on obtient l'image suivante :



On remarque que les chiffres sont globalement reconnaissable bien qu'il doit probablement y avoir une petite erreur sur certaines zones qui ne rendent pas les chiffres complètement "fini" à certaines zones.

Chapitre 6

Question 5-bis

N'ayant pas compris vraiment la question ni comment nous sommes sensé procédé nous avons essayé de répondre à cette question probablement sans succès.

Nous trouvons avec ce calcul un taux d'erreur d'environ 5% (5.35).

Voici le code utilisé.

```
source ("rdfSegmentation.R")

# Chargement de l'image 1
nom <- "rdf-chiffre-1-8bits-seuillage-auto.png"
image1 <- rdfReadGreyImage (nom)

nom <- "rdf-chiffre-1-8bits_classe_a_trouver.png"
image2 <- rdfReadGreyImage (nom)

# Calcul et affichage de son histogramme
nbins <- 256
h1 <- hist (as.vector (image1), freq=FALSE, breaks = seq (0, 1, 1 / nbins))

h2 <- hist (as.vector (image2), freq=FALSE, breaks = seq (0, 1, 1 / nbins))

N <- sum(h1$counts[0:256])
N1 <- h1$counts[0:256]
N2 <- h2$counts[0:256]
print(N)
print(N1)
print(N2)

Erreur <- (h1$density[1]/h2$density[1])+(h1$density[256]/h2$density[256])
print(Erreur)
```