Université de Jijel
Faculté des Sciences et de la Technologie
Département d'Electronique
Master I: ST 18

TP N° 01

Calcul de l'entropie, code uniquement décodable

A) Calcul de l'entropie, calcul des probabilités et des fréquences

1) Ecrire une fonction en Matlab intitulée entrop.m pour calculer l'entropie d'une source (Signal,image,…) dont la syntaxe est: H = entrop(X)
(entrée: l'image X par exemple; sortie: l'entropie H).
Le calcul de l'entropie d'une image en utilisant la fonction «entrop» s'effectue en 3 étapes:

a) Transformer l'image X en un vecteur colonne.

b) Evaluer les probabilités de X en créant une fonction nommée proba.m

c) Calculer l'entropie en utilisant l'expression: $H(X) = -\sum_{i=1}^{N} P(x_i)\ \log_2(P(x_i))$

La fonction «proba» a la syntaxe suivante: **[P,Frequence,Alph]= proba(X)**
Où: Alph: est l'alphabet du vecteur source X.
P: est le vecteur de probabilités correspondant aux lettres de l'alphabet.
Frequence: est le vecteur comportant la fréquence d'occurrence de chaque lettre de l'alphabet.

Etapes à suivre pour créer la fonction «proba()»:

a) Déterminer l'alphabet de X.
b) Calculer les fréquences par 2 méthodes différentes.
c) Estimer les probabilités.

B) Ecrire une fonction nommée unique_decod() qui permet de tester si un code est uniquement décodable: l'entrée: mots de code des symboles. La fonction doit aussi tester si le code est à préfixe.

Université de Jijel
Faculté des Sciences et de la Technologie
Département d'Electronique
Master I: ST 18

TP N° 02

Codage de Huffman

A) Codage de Huffman: fonctions à utiliser: huffmandict(), huffmanenco() et huffmandeco()

Soit une source A={ a_1, a_2, a_3, a_4, a_5} ayant les probabilités {0.2,0.4,0.2,0.1,0.1}.

Afficher les mots de code de chaque symbole, et la longueur moyenne du code de Huffman.

Afficher les mots de code de chaque symbole, et la longueur moyenne du code de Huffman à variance minimale.

B) Calcul de l'entropie par une boucle, de la longueur moyenne et évaluation de l'efficacité d'un code

1) Ecrire une fonction en Matlab intitulée entropie.m pour calculer l'entropie d'une source dont la syntaxe est: H = entropie(p)

Où:p est le vecteur de probabilités correspondant aux lettres de l'alphabet.

2) Ecrire une fonction en Matlab intitulée average_cod.m pour calculer la longueur moyenne du code du Huffman dont la syntaxe est: Lh = average_cod(p,dict)

Où : dict est une cellule dont la 2ème colonne contient les mots de code de chaque symbole. (la 1ère colonne contient les symboles)

3) Evaluer l'efficacité d'un code en utilisant l'expression H/L. (L: longueur moyenne du code concerné (Lh pour Huffman)

B) On considère une source composée de 3 symboles (1,2,3) ayant les probabilités {0.1,0.1,0.8}. la séquence à coder est donnée par l'instruction :

```
sig = repmat([3 3 1 3 3 3 3 3 2 3],1,50);
```

1. Effectuer le codage de Huffman en affichant le mot de code de chaque symbole, et le code de Huffman.

2. Evaluer l'efficacité du code.

3. Comparer la séquence décodée avec l'originale et calculer la longueur moyenne du code par 3 méthodes.

4. Refaire la question **(B)** en supposant qu'on a seulement la séquence à coder. Vous devez déterminer l'alphabet et le vecteur de probabilités correspondant en utilisant la fonction proba() définie précédemment.

C)1. Effectuer le codage de Huffman pour l'image «Cameraman» :(Transformer l'image en un vecteur, effectuer le codage, effectuer le décodage, transformer le vecteur en une image, vérifier la reconstruction)

2. Calculer l'entropie de l'image par la fonction Matlab entropy(). Comparer avec les fonctions entrop() et entropie().

Université de Jijel
Faculté des Sciences et de la Technologie
Département d'Electronique
Master I: ST 18

TP N° 03

Codage arithmétique, Comparaison entre le codage arithmétique et le codage de Huffman,le codage à base du dictionnaire (LZW), codage RLE

I) Codage arithmétique

A) Ecrire un programme en Matlab qui permet d'implémenter la méthode du codage arithmétique, exposée au cours du module ST15.

B) En utilisant les fonctions Matlab arithenco() et arithdeco()effectuer le codage arithmétique de la séquence: sig = repmat([3 3 1 3 3 3 3 3 2 3],1,50);

1.Comparer la séquence décodée avec l'originale et calculer la longueur moyenne du code (La), et le taux de compression Tca.

2.Evaluer l'efficacité du code.

3.Comparer le résultat obtenu avec le codage de Huffman (question **(B.2)** du TP 02)

4.Refaire la question **(B)** en supposant qu'on a seulement la séquence à coder. Vous devez déterminer l'alphabet et le vecteur de fréquences correspondant en utilisant la fonction proba() définie précédemment.

C) Refaire la question **(B)** en supposant que la séquence originale contient 10000 symboles. Commenter. Vous devez avoir l'exécution suivante:

La taille de la séquence 1 (avant codage) est: ? bits. 500 symboles
Lh1= ? bits/symb, H1=? bits/symb, zh1=?, Tch1=?. Codage de Huffman pour 500 symboles
La taille de la séquence 2 (avant codage) est: ? bits. 10000 symboles
Lh2= ? bits/symb, H2=? bits/symb, zh2=?, Tch2=?. Codage de Huffman pour 10000 symboles
La taille de la séquence 1 (avant codage) est: ? bits. 500 symboles
La1= ? bits/symb, H1=? bits/symb, za1=?, Tca1=?. Codage arithmétique pour 500 symboles
La taille de la séquence 2 (avant codage) est: ? bits. 10000 symboles
La2= ? bits/symb, H2=? bits/symb, za2=?, Tca2=?. Codage arithmétique pour 10000 symboles

E) Effectuer le codage arithmétique pour l'image «cameraman» :(Transformer l'image en un vecteur, effectuer le codage, effectuer le décodage, transformer le vecteur en une image, vérifier la reconstruction).

1. Calculer le taux de compression et le débit.

2. Comparer le résultat obtenu avec le codage de Huffman.

3. Mesurer le temps de codage et décodage pour les 2 méthodes.

II) Codage à base du dictionnaire (L'Algorithme Lempel-Ziv-Welch:LZW) et codage RLE

Ecrire un programme en Matlab qui permet d'implémenter les méthodes du codage LZW et RLE.

Université de Jijel
Faculté des Sciences et de la Technologie
Département d'Electronique
Master I: ST 18

TP N° 04

Quantification scalaire uniforme, compression d'images avec perte, codage prédictif, compression sans perte

I) Quantification scalaire uniforme, compression d'images avec perte, et évaluation de la qualité des images décompressées

1. Ecrire un programme en Matlab qui permet de concevoir un quantificateur scalaire uniforme d'une image en supposant que les pixels varient uniformément entre 0 et 255 avec des taux de 1 bit, 2 bits et 3 bits.

2. Appliquer le codage de Huffman aux images originales et quantifiées (codage et décodage)en calculant l'entropie et le débit. Que constater ?

3. Evaluer la qualité des images décompressés en calculant le PSNR ou le SSIM.

II) Codage prédictif et compression sans perte

1. On considère la séquence sig=[1 2 3 2 3 4 5 4 5 6 7 8 9 8 9 10]

a. Calculer l'entropie de cette séquence.

b. Appliquer le codage de Huffman et calculer la longueur moyenne.

2. Créer deux fonctions pred.m et pred_inv.m pour le codage prédictif en utilisant les équation suivantes respectivement:

Codage prédictif: $r_n = x_n - x_{n-1}$

Codage prédictif inverse: $x_n = x_{n-1} + r_n$

3. Appliquer le codage de Huffman à la séquence résultante de l'application du codage prédictif à la séquence sig. Calculer l'entropie et la longueur moyenne. Qu'observe-t-on ?

4. Refaire la question **3)** pour:

a. La séquence: sig2=repmat([1 2 3 2 3 4 5 4 5 6 7 8 9 8 9 10],1,50)

b. la première ligne de l'image cameraman.

c. l'image cameraman

Comparez les performances avec la technique du codage de Huffman appliqué sans codage prédictif. Qu'observe-t-on ?