

---

# TP02 - Codage de Huffman

---

Rémi BURTIN

Cyril FOUGERAY

30 mars 2014



UNIVERSITÉ DE TECHNOLOGIE DE  
COMPIÈGNE

# 1 Introduction

L'objectif de ce TP est de comparer le codage de Huffman au codage arithmétique. Pour cela, nous travaillons sur une image ('lena.bmp') en niveaux de gris, ce qui permet de fixer une seule valeur (entre 0 et 255) pour chaque pixel de l'image. Depuis les probabilités des pixels, que nous avons calculé via un histogramme (cf. TP précédent), nous avons implémenté les deux algorithmes, puis calculé le taux de compression de chacun.

## 2 Codeur de Huffman

Afin de réaliser l'encodage, il est nécessaire de connaître la distribution de probabilités des niveaux de gris de l'image. Pour cela, nous commençons par réaliser l'histogramme comptant le nombre d'occurrences de chaque valeur de gris.

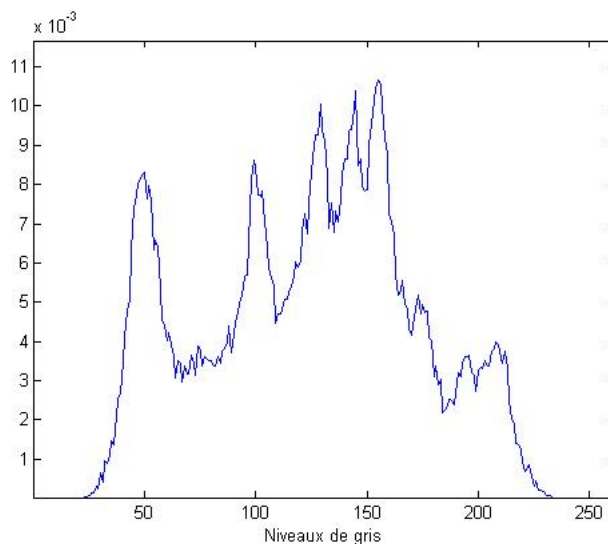


FIGURE 1 – Distribution de probabilités, niveaux de gris

## 3 Codeur arithmétique

## 4 Conclusion

# A Codes source MATLAB

## A.1 Algorithme - codage de Huffman

```
1 function compression = huffman(path)
2 global prob avglen dict;
3
4 % matrice de l'image
5 disp(['Lecture de l''image ' path]);
6 img = imread(path);
7
8 % initialisation vecteurs
9 histogramme = zeros(1,256); % occurrences par niveaux de gris
10 prob = zeros(1,256); % probas (histo normalise)
11
12 composantes = size(img, 3); % Taille de la 3eme dimension
13
14 if (composantes == 3) % conversion en gris si image couleur
15     disp('Image couleur : conversion en gris');
16     img = rgb2gray(img);
17 end
18
19 dim = size(img,1)*size(img,2); % nombre de pixels dans l'image
20 list=reshape(double(img), 1, dim); % images en ligne (vecteur)
21
22 % creation histogramme
23 for i=1:dim,
24     histogramme(round(list(i))) = histogramme(round(list(i))) + 1;
25 end
26
27 % creation probas des niveaux de gris
28 disp('Creation vecteur des probalites des niveaux de gris');
29 for i=1:256,
30     prob(i) = histogramme(i)/dim;
31 end
32
33 % creation vecteur de taille 256 (1, 2, 3, ... 256)
34 % correspond aux niveaux de gris
35 disp('Creation du vecteurs des differents symboles');
36 symbols = [1:256];
37
38 % creation du dictionnaire
39 disp('Creation du dictionnaire (via huffmandict())');
40 [dict,avglen] = huffmandict(symbols,prob);
41 disp(['Longueur moyenne des mots encodes : ' num2str(avglen)])
42
43 % encodage de l'image (passer un vecteur ligne)
44 disp('Encodage de l''image');
45 tic % calcul du temps d'encodage/decodage
46 enco = huffmanenco (list, dict);
47
48 % decodage (retourne un vecteur)
49 disp('Decodage de l''image');
50 deco = huffmandeco(enco, dict);
51 time = toc;
```

```

52 disp(['Duree encodage/decodage : ' num2str(time) 's.']);
53
54 %reshape vecteur -> image 2D
55 imagedeco = reshape(deco, size(img,1), size(img,2));
56
57 imshow(imagedeco/256);
58
59 compression = 1-length(enco)/(dim*8);
60 disp(['Taux de compression : 1 - taille finale / taille initiale = ' num2str(compression)]);
61
62 compression2 = 1-avglen/8;
63 disp(['Taux de compression : 1 - longueur moyenne mot code / 8 = ' num2str(compression2)]);
64
65 return;
66 end

```