

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique

Université des Sciences et de la Technologie  
HOUARI BOUMEDIENE

B. P. 32, El-Alia, 16111 Bab-Ezzouar, ALGER  
Téléphone/Fax: +213 21 24 76 07



الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي  
والبحث العلمي

جامعة هواري بومدين  
للعلوم والتكنولوجيا

ص. ب. 32، العليا، 16111، باب الزوار، الجزائر  
الهاتف / الفاكس : +213 21 24 76 07

Année 2017/2018

Master Informatique Visuelle  
Communication Multimédia

### Série de travaux pratiques n°2

#### Mise en œuvre d'un Codec RLE pour les images binaires (cas du fax)

##### Exercice 1

Il s'agit de mettre en œuvre un codec (codeur/décodeur) basée sur la technique RLE.

Un pixel a deux valeurs possibles (0 pour le noir et 255 pour le blanc) et donc codé sur un octet : 00H ou FFH.

Une image de  $n$  pixels sera donc représentée par une chaîne de  $n$  caractères. Il s'agit de compresser et de décompresser une telle image. Pour cela, le codage suivant est utilisé :

- Si le même pixel se répète 3 fois ou plus, on utilisera un mot de deux octets : le bit de poids fort est à 1, et les 15 autres bits servent à coder le nombre de répétitions du pixel. Ces deux octets sont ensuite suivis de la couleur répétée.
- Si on a une suite de pixels différents (ou qui ne se répètent pas plus de 2 fois), on fera précéder cette suite d'un mot de deux octets où le bit de poids fort est à 0, et les 15 autres bits servent à coder la taille de la suite.

1- Ecrire le programme correspondant en C++ qui permet de lire une image et produit un fichier de codes en hexadécimal représentant l'image compressée. Appliquer le programme pour les deux cas d'images de fax fournies (noir et blanc). Calculer les taux de compression pour chacune des images.

2- Ecrire le programme correspondant en C qui permet de lire un fichier compressé et de retrouver l'image d'origine et de la visualiser.

A titre d'exemple, les codes obtenus par cette méthode pour les images de la figure 1 sont les suivants :

- a) (0024H) 00H FFH 00H FFH ....00H FFH, soit  $2+36=38$  octets
- b) (8003H) 00H (8006H) FFH (8006H) 00H (8003H) FFH (8003H) 00H (8006H) FFH (8006H) 00H (8003H) FFH, soit  $3 \times 8=24$  octets
- c) (800CH) 00H (800CH) FFH (800CH) 00H, soit 9 octets

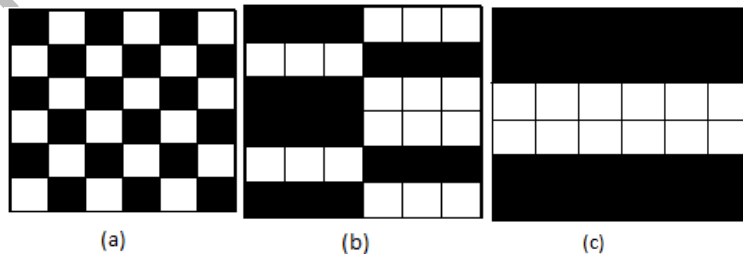


Figure 1. Exemple d'images