**Master I : IL 10 mai 2010**

**Enseignant :Abdelli**

**Contrôle N.1 : Systèmes Multimédia**

1. Soit l’Image suivante composée de six zones (a,b,c,d,e f) colorées chacune avec une teinte unie. Donnez les composantes Rouge, Vert, Bleu, magenta pour cette image.

13 cm

|  |  |  |
| --- | --- | --- |
| **A :** Rouge  3cm | **B :** Cyan | **C : Blanc** |
| **D :** noir | **E**: Jaune | F : Bleu |

1. Supposons que la résolution de l’image précédente est de (50 Pixels par Cm) donnez son poids dans les cas suivants :
   1. Codage RVB.
   2. 256 Couleurs indexées.
2. Si un algorithme de compression de type RLE est utilisé, quel serait approximativement son poids dans le cas (b) ; expliquez la démarche.
3. Appliquez l’algorithme de Huffman pour compresser le fichier texte suivant : « *Je souhaite avoir une bonne note dans le module de systèmes multimédia* ». estimez la taille du fichier compressé.
4. Quel est l intérêt d’utiliser le modèle YUV par l’algorithme JPEG, et quel est le but d’appliquer la transformée DCT sur la composante **Y**.
5. Soit une séquence audio de 2 minutes, estimez la taille du fichier dans les cas d’une numérisation avec :
   1. Une qualité téléphone.
   2. Une qualité CD audio.

**Nom  et Prénom : Matricule : signature**

**Réponse :**

1. Les composantes Rouge vert et bleu sont utilisées dans le modèle RVB, la composante magenta dans le modèle CMJ.

**Composante rouge : A, C,E🡪 rouge ; B,D,F🡪 noir.**

**Composante Verte : B, C,E🡪 verte ; A,D,F🡪 noir.**

**Composante Bleu : B,C,F🡪 bleu  ; A,D,E🡪 noir.**

**Composante Magenta : A,D,F 🡪 Magenta  ; B,C,E🡪 blanc.**

1. Résolution =50 Pixels par cm 🡺 Définition = (13\*50) \* (3\*50)
2. **Codage RVB.**

Poids = Définition \* Nombre d’octet par pixel =(13\*50) \* (3\*50)\*3 = **292 500 octets**

**256 Couleurs indexées.**Poids = Définition \* Nombre d’octet par pixel =(13\*50) \* (3\*50)\*1 = **97 500 octets**

1. L’algorithme RLE élimine compresse en éliminant les répétitions des codes consécutifs en explorant l’image dans le sens horizontal. Chaque ligne de codes de Pixels contigus de même couleur sont réduits par le code de la couleur suivi du nombre de répétitions. Donc chaque bloc (A,B, C ,D ,E, F) sera réduit à deux colonnes de codes ; on obtient dans le cas indexé :

Poids RLE = Nbre de blocs \* (Longueur du bloc \* Résolution\* 2 colonnes \* taille code)

Poids RLE = 6 \* (1,5 \* 50\* 2 \* 1 octet) = **900 octets**

1. Taille fichier non compressé = 69 caractères= **69 octets**

On calcule les itérations de chaque lettre dans la phrase :

A (4), b(1), d(3), e(11), h(1), i(3), j(1), L(3), m(4), n(5), o(5) , r(1), s(5),t(4),u(4), v(1), y(1), espace (11)

Je construis ensuite l’arbre de huffman, et j’attribue mes nouveaux codes.

J’estime la taille de la table de huffman. Puis la taille du texte compressé. Je calcule la somme pour obtenir la taille du fichier compressé. Je calcule le rapport de compression.

1. Le passage vers le modèle YUV permet de traiter la composante Y (luminance) différemment des composantes de chrominance rouge et bleu U et V car l’œil humain se sensibilise plus aux variations de lumières qu’aux variations de couleurs.

L’application de la DCT sur la composante Y permet d’avoir une représentation par moyennes ç.a d : estimer le coefficient de la moyenne d’un bloc et les coefficients des écarts à partir de cette moyenne ; ceci permet d’envisager de réduire les coefficients non significatifs à zéro après quantification.

1. 2 minutes = 120 secondes
   1. Qualité téléphone : 8000 Hz \_ 1 octet \_ Mono

Taille fichier = 120\* 8000\* 1\*1= **960 000 octets**

* 1. Qualité téléphone : 44 100 Hz \_ 2 octet \_ Stereo

Taille fichier = 120\* 44 100\* 2\*2= **21 168 000  octets**