Examen Codage Transmission Compression M2IM 2004-2005

Le 2/2/2005, Tout document autorisé

Recopier tel quel des extraits de documents n'apporte jamais de points. Vous devez répondre aux questions qui vous sont posées. Vous devez argumenter (étayer) vos réponses. Une réponse correcte non étayée ne rapporte pas de points, et une réponse incorrecte bien étayée peut en rapporter.

Questions de cours

1. Donnez quelques méthodes pour générer des fichiers non compressibles.
2. Les fichiers qui contiennent des programmes exécutables sont-ils compressibles ?
3. Que ce passe-t-il si l'on applique un algorithme de compression de son à une image (qualité, taux de compression, ...) ?
4. Pourquoi les compresseurs ajoutent souvent un CRC ?
5. Citez des cas ou le *Block Sorting* n'est pas adapté.
6. Quel est le rapport entre le traitement d'image et MPEG4 ?
7. Que faut-il faire pour compresser en JPEG sans pertes ?
8. Citez les différents avantages différents apportés par l'utilisation de la compression.
9. Comment faites-vous pour savoir si un support de sauvegarde compresse les données à votre insu ?
10. Faut-il compresser une archive (`tar`) ou faire une archive de fichiers compressés ?

Exercices

11. Vous avez le droit d'utiliser tous les outils existants sur les ordinateurs. Comment vous y prendriez-vous pour transférer avec le plus fort taux de compression une image carrée contenant le plus grand disque possible blanc sur fond noir ? Votre solution donne-t-elle la plus fort taux de compression quelque soit la taille initiale de l'image ?
12. Prenez les 20 premiers caractères de votre "Prénom Nom Prénom Nom". Donnez les codages de SHANNON FANO et HUFFMANN. Indiquez correctement toutes les informations qui permettront une correction facile de l'exercice.
13. Le RLE parcourt l'image ligne à ligne, comment parcourir l'image pour que cette technique donne de meilleurs résultats ?
14. Proposez une méthode pour stocker des milliards de résultats du loto en prenant le moins de place (7 boules numérotés de 1 à 49). Votre méthode est-elle optimale en temps et en place ?
15. Comparez la complexité en temps de la compression DCT et ondelettes dans un espace à 1, 2 ou 3 dimensions. Attention, cette question est intentionnellement imprécise.

Corrigé

1. Une fichier de contenu aléatoire, un fichier comprimé

2. Il n'y a rien d'aléatoire dans un code exécutable, il est donc comprimable. Par contre un compresseur de texte "normaux" a du mal à retrouver les événements.

3. Moins bonne compression car ne tient pas compte que l'image est un signal 2D. Beaucoup de pertes visibles car la psycho-acoustique ne s'applique pas à des images. Évidement, la psycho-acoustique ne supprime pas les hautes fréquences.

4. L'utilisation de CRC est fondamentale car par exemple un fichier C comprimé qui a quelques bits modifiées pourra donner après décompression un fichier C faux mais qui a de fortes chances de passer à la compilation.

5. Les block sorting n'est pas adapté quand la compression doit être interactive car elle ne peut commencé que quand on a assez de données.

6. Quand on comprime en MPEG4 on sépare l'image en différentes zones pour comprimer séparément ces zones. Il faut faire du traitement d'image pour trouver ces zones.

7. Avec la meilleur qualité des erreurs sont du au fait que les coefficients de la DCT sont stockés dans des entiers. Pour transmettre sans erreur il est donc nécessaire de stocker les erreurs commises.

8. On vous annonce une taille de X pour votre support de sauvegarde. Si vous pouvez sauver un fichier aléatoire de taille X alors il n'y a pas de compression à votre insu.

9. Comprimer un "tar", avantages : plus facile de trouver de la redondance, inconvénients : pas facile de compresser ensemble des choux et des carottes. Un choix astucieux est de trier les fichiers par type avant de les mettre dans l'archive afin que les fichiers se ressemblant soient comprimés ensemble.

10. On transfert un programme C dont l'exécution fait l'image que l'on désire transférer. C'est la meilleur compression pour les images dont la taille après compression est supérieure à la taille du programme C après compression. Pour cet exercice, l'expérience à montré que cet la meilleur compression pour des images plus grande que 51x51 pixels.

11. Il était demandé de donner toutes les informations pour que le correcteur puisse vérifier votre travail. Donc notamment les arbres.

12. Parcourir suivant la courbe de Peano (cherchez sur le web).

13. Quelques possibilités :

    - 6 bits par numéro de boule : 42 bits par tirage, trivial à coder et à accéder.
    - Codage du tirage comme un nombre en base 49 : le plus grand nombre est 678 223 072 849, il faut 39.30 bits pour le coder. Si l'on accepte de perdre 0.7bits par tirage le codage est trivial et l'accès directe. Sinon la formule est : $b_1^1 + 49(b_2^1 + 48(b_3^1 + 47 * (b_4^1 + 46(b_5^1 + 45(b_6^1 + 44(b_7^1 + 43(b_1^2 + 49(b_2^2 + 48(b_3^2 + 47 * (b_4^2 + 46(b_5^2 + 45(b_6^2 + 44(b_7^2 ...)))))))))))))$ La place occupée est minimale mais le temps d'accès est prohibitif.

- Codage du tirage comme un nombre n'indiquant pas l'ordre des boules. Le nombre de tirage différents est : $\frac{43\ 44\ 45\ 46\ 47\ 48\ 49}{7!} = 85900584$ et cela ce code sur 26.3 bits.
  On suppose accepter la perte de 0.7 bits par tirage. Dans ce cas le codage est complètement différent car il dépend des valeurs numériques, on suppose que les numéros de boules sont triés par ordre croissant. Si le plus petit numéro est 44 on a pas besoin d'indiquer les autres, dans ce cas, dans la formule suivante il y a des multiplications par 0. **Cette formule n'atteint pas l'optimum. Je suis preneur si quelqu'un trouve une formule optimum** $b_1 + (49-6)(b_2 - b_1 - 1 + (49-5-b_1-1)(b_3 - b_2 - 1 + (49-4-b_2-1)(...)))$

14. n est la taille du coté, pas le nombre de cellules.

Dimension	Temps DCT	Temps ondelette
1	$O(n^2)$	$O(n\log n)$
2	$O(n^3)$	$O(n^2\log n)$
3	$O(n^4)$	$O(n^3\log n)$