

Transformée de Fourier discrète appliquée à la compression de données

26 février 2017

1 Positionnement thématique

Mathématiques (analyse), Informatique (informatique pratique), Mathématiques (autre)

2 Mots clés

Compression (*compression*)
Echantillonnage (*sampling*)
Transformée de Fourier discrète (*discrete Fourier Transform*)
Transformée de Fourier rapide (*Fast Fourier transform*)
Codage entropique (*entropy coding*)

3 Bibliographie commentée

L'objectif de la compression de données est de réduire significativement la taille de fichiers (images, vidéos, fichiers audio) afin de pouvoir en stocker un maximum en utilisant un minimum de mémoire. La taille du fichier est le nombre d'octets (groupes de 8 bits) codant l'information, on évalue la qualité de la compression avec le taux de compression $t = 1 - i/f$ (en %) , où i et f sont les tailles respectives du fichier avant et après compression. Il s'agit donc de maximiser ce taux.

La compression est une opération informatique : elle ne s'applique qu'à des signaux numériques (discrets). Il est donc nécessaire pour traiter un signal analogique (continu) de l'échantillonner, c'est-à-dire de le représenter sous la forme d'un signal numérique en mesurant un nombre fini de valeurs. Se pose alors la question de la fréquence d'échantillonnage : à quelle fréquence faut-il effectuer les mesures pour obtenir une représentation exacte d'un signal analogique ? Claude Shannon, qui a développé dans les années 1940 la théorie de l'information, a démontré que la fréquence d'échantillonnage doit être au moins égale au double de la largeur du spectre du signal (le spectre étant l'ensemble des fréquences présentes dans la décomposition en série de Fourier d'un signal) : c'est le théorème de Shannon [1][2].

Il existe des procédés de compression sans pertes, qui conservent exactement les informations du fichier initial, mais ils sont limités. Pour obtenir des taux de compression importants, on s'autorise alors des pertes d'information. Il faut alors veiller à ne pas trop dégrader la qualité du fichier : pour cela, on ne supprime que des composantes du signal jugées négligeables, car imperceptibles ou presque par l'oeil ou l'oreille humaine. C'est là qu'intervient la transformée de Fourier discrète [2] : elle permet d'étudier un signal échantillonné dans le domaine fréquentiel, en en donnant un spectre discret. On supprimera ainsi des fréquences du spectre (et non des valeurs prises par le signal), ce qui est efficace car le cerveau humain ne perçoit pas les contrastes trop faibles ou les fréquences trop hautes ou basses. Par exemple, les fréquences audibles sont comprises entre 20Hz et 20kHz environ, les autres fréquences sont donc négligeables : on ne perçoit pas de baisse de qualité si on les supprime.

En plus de permettre le passage du domaine temporel au domaine fréquentiel, la Transformée de Fourier discrète est un outil adapté à la compression car elle admet une réciproque, qui est aussi une Transformée de Fourier discrète : on peut ainsi facilement récupérer un signal temporel à partir de son spectre, et il est

donc possible de représenter un signal uniquement par ses fréquences sans rendre difficile la décompression. De plus, l'algorithme FFT (Fast Fourier transform), publié par James Cooley et John Tuckey en 1965 [3][4], permet de calculer efficacement la transformée de Fourier discrète (en $O(n \lg(n))$ pour n points), et donc de compresser et décompresser rapidement des fichiers.

Les algorithmes de compression combinent traitement fréquentiel du signal et compression sans pertes : la compression par Transformée de Fourier discrète induit de nombreuses répétitions, ce qui maximise l'efficacité d'un codage entropique, qui consiste à coder les caractères les plus fréquents avec moins de bits que les plus fréquents (au lieu du code ASCII sur 7 ou 8 bits par caractère), comme le codage de Huffman [5][6], publié en 1952, qui crée un nouveau code à l'aide d'arbres binaires. Ainsi, l'algorithme JPEG [7] de compression d'images utilise une forme bidimensionnelle de transformée de Fourier discrète [8], la transformée en cosinus discrète, puis une compression sans pertes, dont le principal élément est un codage de Huffman.

4 Problématique retenue

Comment stocker un signal en utilisant un minimum de mémoire tout en conservant la qualité du fichier ? Il s'agit d'étudier les différents outils permettant de compresser un fichier et de s'interroger sur les taux de compression atteignables sans (ou presque sans) le dégrader.

5 Objectifs du TIPE

Etude théorique de l'échantillonnage du signal et des résultats d'analyse harmonique utiles à la compression de fichiers.

Etude de l'algorithme JPEG, puis implémentation de l'algorithme FFT et d'un algorithme de compression d'images et analyse des résultats obtenus.

J'étudierai plus particulièrement les algorithmes utilisés dans notre procédure de compression, dans le but de les optimiser. J'implémenterai notamment la transformée de Burrows-Wheeler [9], qui réarrange les données sans les compresser, mais favorise le codage entropique.

6 Références bibliographiques

- [1] Claude Shannon, Communication in the presence of noise, 1949
<http://web.stanford.edu/class/ee104/shannonpaper.pdf>
- [2] Cours de G. Baudoin et J.F. Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique
https://perso.esiee.fr/~bercherj/New/polys/g_signal.pdf
- [3] James Cooley et John Tukey, An algorithm for the machine calculation of complex Fourier series <http://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf>
- [4] Thomas Cormen, Charles Leiserson, Ronald Rivest et Clifford Stein, Algorithmique, Dunod, 2002
- [5] D.A. Huffman, A method for the construction of minimum-redundancy codes, 1952
http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf
- [6] Wikipédia : Codage de Huffman
https://fr.wikipedia.org/wiki/Codage_de_Huffman
- [7] Cours sur JPEG
<http://www.chireux.fr/mp/cours/Compression%20JPEG.pdf>
- [8] Wikipédia : Transformée en cosinus discrète
https://fr.wikipedia.org/wiki/Transform%C3%A9e_en_cosinus_discr%C3%A8te
- [9] Michael Burrows et D.J. Wheeler, A block-sorting lossless data compression algorithm, 1994
<http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.pdf>