

Compression de données

On peut distinguer 2 types de compression :

La compression destructive permet de diminuer la taille des données en supprimant des éléments inutiles ou invisibles. Exemple : Compression d'image en JPEG ou de musique en MP3.

Le taux de compression peut être réglable et être très élevé au prix d'une perte de qualité de la donnée.

La compression non destructive permet de diminuer la taille des données en éliminant les éléments redondants ou en réorganisant les données. Il existe une limite théorique au taux de compression maximal.

Compression Simple : RLE

Run-Lenght Encoding

Cf http://fr.wikipedia.org/wiki/Run-length_encoding

Exemple avec une série de valeurs :

5 5 6 8 1 8 8 8 8 8 8 8 8 6 6 6 6 1 5

On défini un code spécial pour indiquer qu'une valeur est répétée. Par exemple le code « A ». Si on voit une valeur se répéter plus de 2 fois, on indique « A » puis la valeur puis le nombre de répétitions.

5 5 6 8 1 A 8 9 A 6 4 1 5

Par contre la valeur « A » ne doit pas être présente dans la série de valeur à compresser. Sinon on peut la représenter comme ceci : A A 1 (mais on utilise 3 valeurs au lieu de 1).

Le résultat de la compression peut quelquefois être plus gros que l'original.

Le format RLE est utilisé pour les images « Bitmap » (BMP).

Compression de Huffman

Le codage de Huffman est un algorithme de compression de données sans perte élaboré par David Albert Huffman, lors de sa thèse de doctorat au MIT.

L'algorithme a été publié en 1952 dans l'article A Method for the Construction of Minimum-Redundancy Codes, dans les Proceedings of the Institute of Radio Engineers¹.

Le codage de Huffman utilise un code à longueur variable pour représenter un symbole de la source (par exemple un caractère dans un fichier).

Le code est déterminé à partir d'une estimation des probabilités d'apparition des symboles de source, un code court étant associé aux symboles de source les plus fréquents. Les codes de Huffman sont des codes optimaux, au sens de la plus courte longueur.

Exemple : Le scrabble. La valeur des lettres dépend de leur fréquence dans la langue française. Le « e » vaut 1.

<http://www.commentcamarche.net/contents/video/huffman.php3>

Compression par dictionnaire

LZW (pour Lempel-Ziv-Welch) est un algorithme de compression de données sans perte. Il s'agit d'une amélioration de l'algorithme LZ78 inventé par Abraham Lempel et Jacob Ziv en 1978. LZW fut créé en 1984 par Terry Welch, d'où son nom.

L'algorithme a été conçu de manière à être rapide à implémenter, mais n'est la plupart du temps pas optimal car il effectue une analyse limitée des données à compresser.

Il est utilisé dans les modems (norme V42 bis) et dans les formats d'image numérique GIF ou TIFF et les fichiers audio MOD.

<http://www.commentcamarche.net/contents/video/lzw.php3>

Compression destructive : MP3

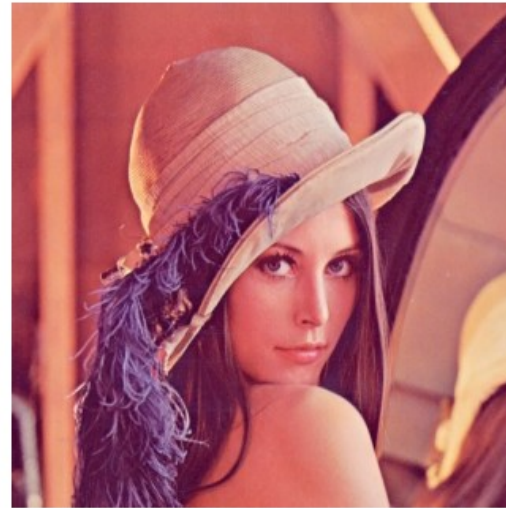
<http://gabriel.braun.free.fr/CoursInfo/Compression/mp3.html>

Compression destructive : JPEG

http://www.iut-arles.up.univ-mrs.fr/thon/LP/IN_recode/LP%20-%20IN%20RepCod%20-%20Chapitre%202.pdf



Original



JPEG

Image 512x512 pixels, 16 millions de couleurs :

- Brut : 769 Ko
- LZW: 631 Ko (ratio = 1.2 : 1)
- JPEG : 52 Ko (qualité 75 %) (ratio = 15 : 1)

Performances de la compression JPEG



Original (769 Ko)



JPEG 75% (52 Ko)

Ratio = 15:1

Performances de la compression JPEG



Original (769 Ko)



JPEG 50% (24 Ko)

Ratio = 32:1

Performances de la compression JPEG



Original (769 Ko)



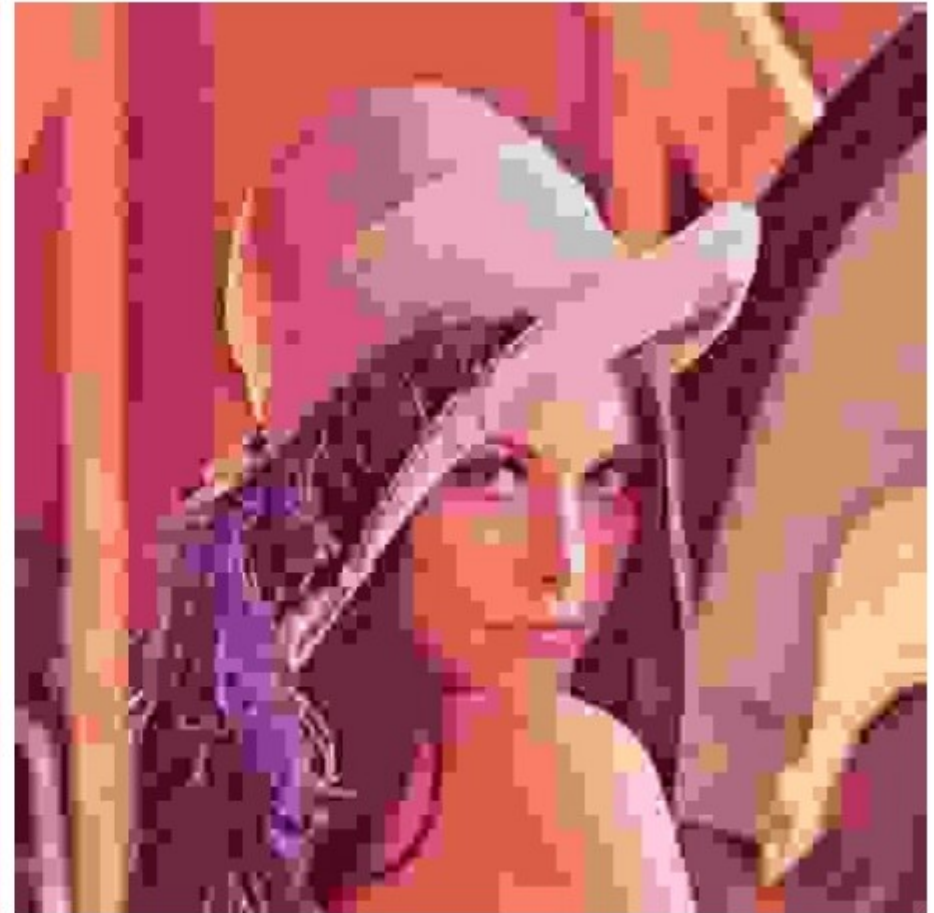
JPEG 25% (16 Ko)

Ratio = 48:1

Performances de la compression JPEG



Original (769 Ko)



JPEG 1% (6 Ko)

Ratio = 128:1

Compression destructive : JPEG

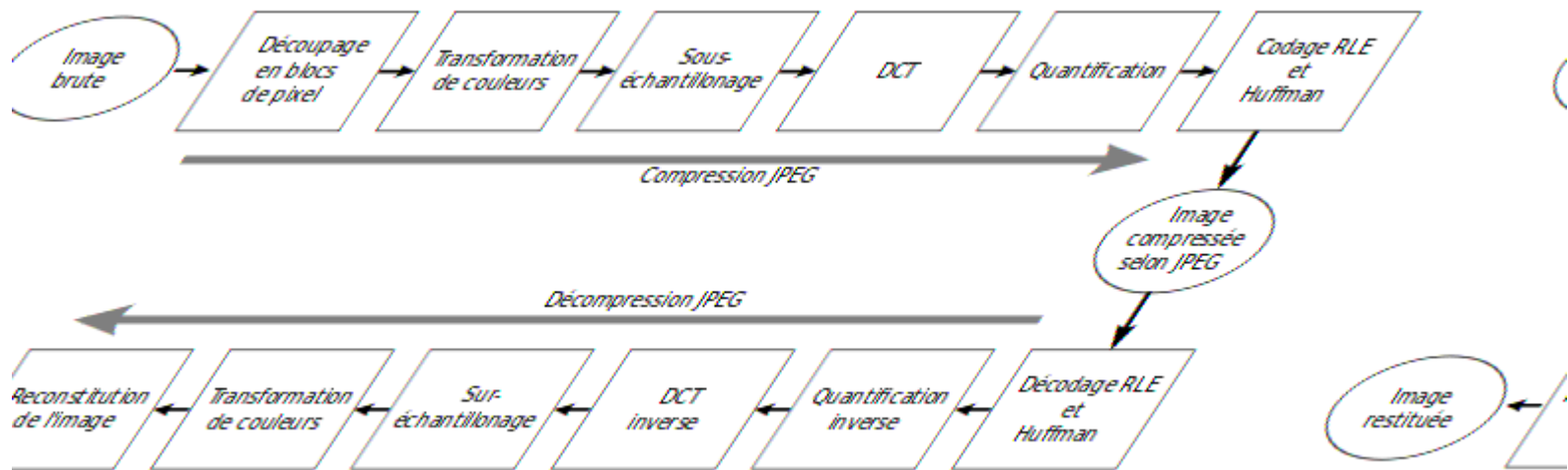
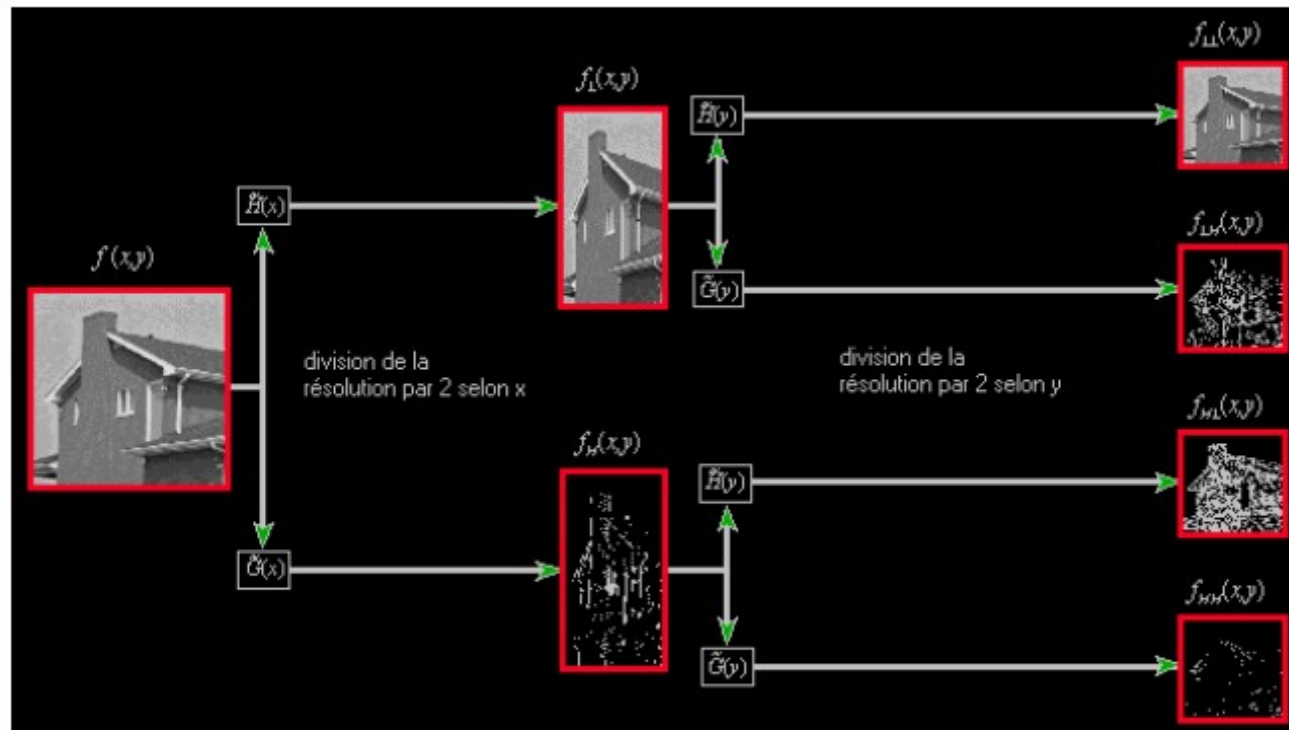
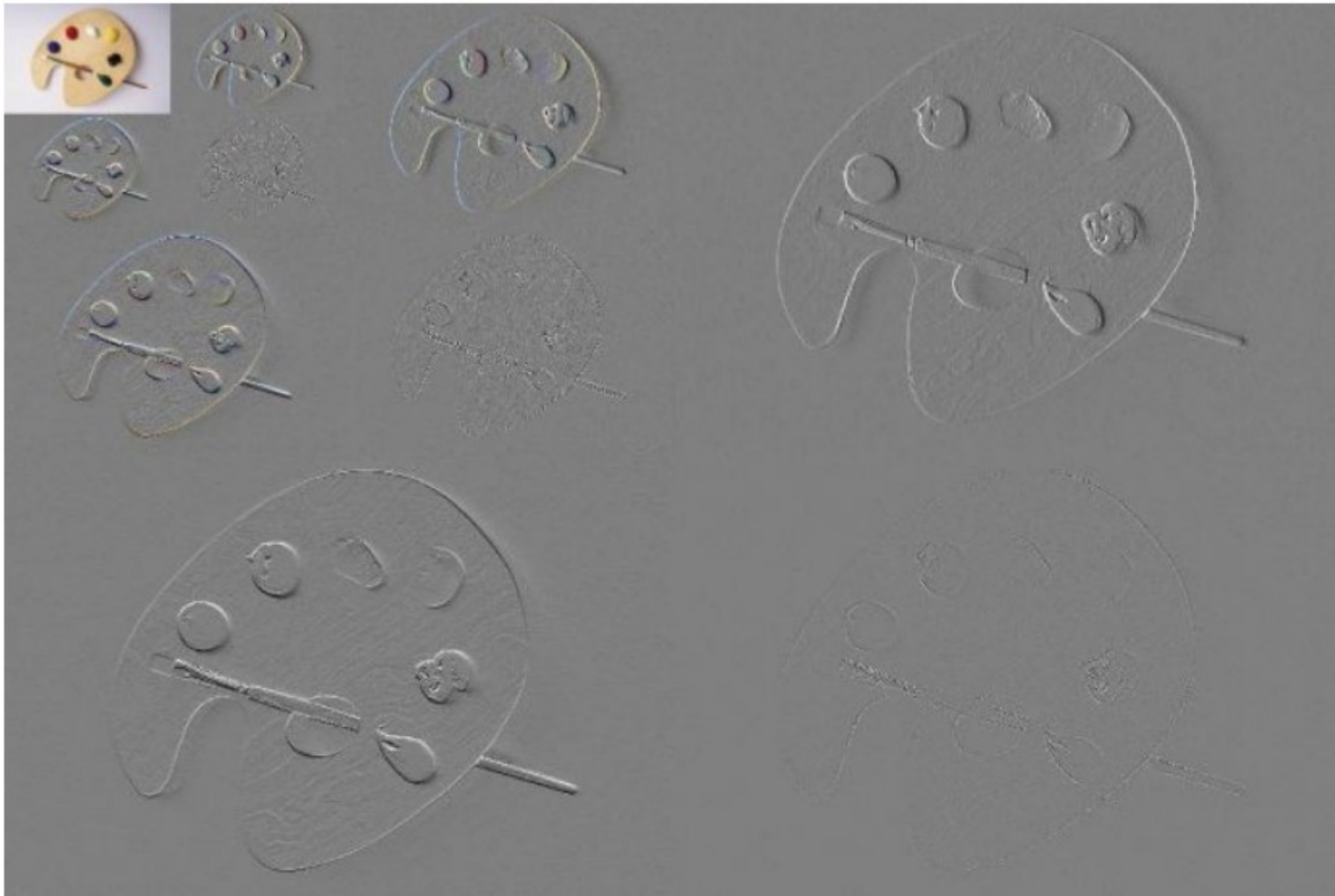


Figure 1 : Organigramme de compression.

Compression destructive : Ondelettes





Structure de données après compression (3 itérations)



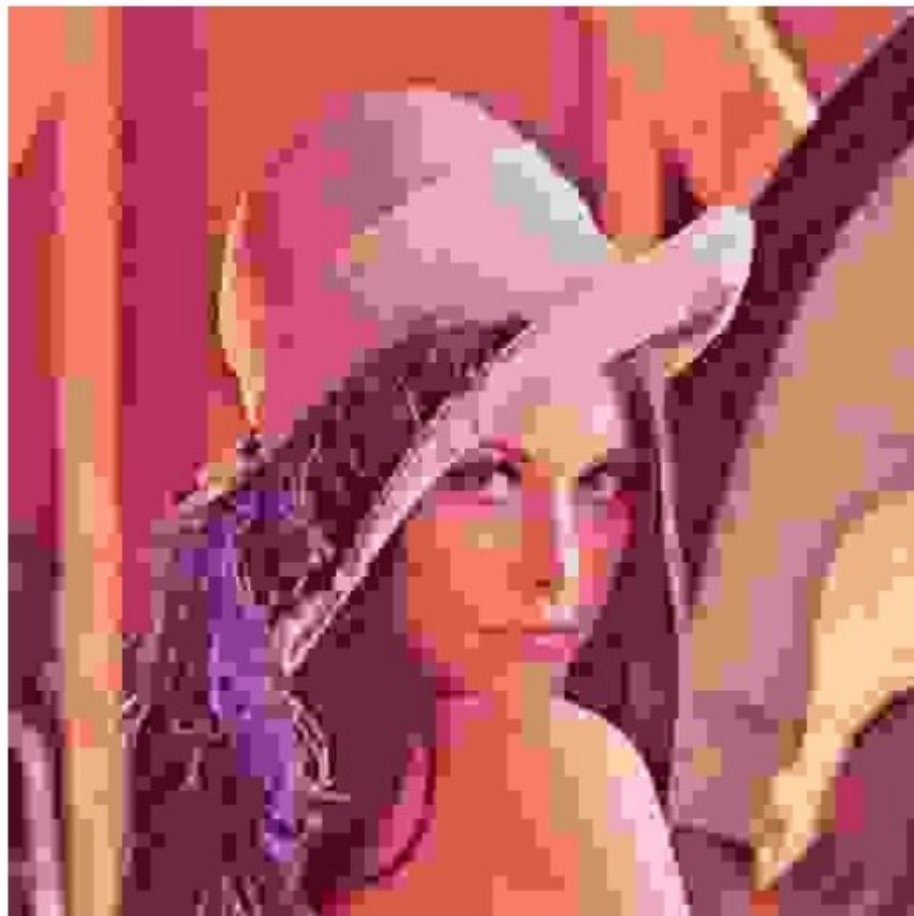
JPEG (29 Ko)

Ratio = 26:1



JPEG 2000 (29 Ko)

Ratio = 26:1



JPEG (6 Ko)

Ratio = 128:1



JPEG 2000 (6 Ko)

Ratio = 128:1